

UNIVERSITE D'AIX-MARSEILLE

ECOLE DOCTORALE EN MATHEMATIQUES ET
INFORMATIQUE DE MARSEILLE (E.D. 184)

FACULTE DES SCIENCES ET TECHNIQUES

LABORATOIRE LSIS UMR 7296

THESE DE DOCTORAT

Spécialité : Informatique

Présentée par :

Shereen ALBITAR

**De l'usage de la sémantique dans la classification
supervisée de textes : application au domaine médical**

*On the use of semantics in supervised text classification: application
in the medical domain*

Soutenue le : 12/12/2013

Composition du Jury :

Pr. Sylvie CALABRETTO	LIRIS-INSA, Lyon	<i>Rapporteur</i>
Pr. Lynda TAMINE	Université Paul Sabatier, Toulouse	<i>Rapporteur</i>
Pr. Nadine CULLOT	Université de Bourgogne, Dijon	<i>Examineur</i>
MCF-HDR. Jean-Pierre CHEVALLET	Université Pierre Mendès France, Grenoble	<i>Examineur</i>
Pr. Patrice BELLOT	LSIS, Université d'Aix-Marseille	<i>Examineur</i>
Pr. Bernard ESPINASSE	LSIS, Université d'Aix-Marseille	<i>Directeur de thèse</i>
MCF. Sébastien FOURNIER	LSIS, Université d'Aix-Marseille	<i>Co-directeur de thèse</i>

ABSTRACT.

Facing the exploding increase in electronic text documents on the internet, it has become a compelling necessity to develop approaches for effective automatic text classification based on supervised learning. Most text classification techniques use Bag Of Words (BOW) model for text representation in the vector space. This model has three major weak points: Synonyms are considered as distinct features, polysemous words are considered as identical features keeping ambiguities unresolved. In fact, these weak points are essentially related to the lack of semantics in the BOW-based text representation. Moreover, certain classification techniques in the vector space use similarity measures as a prediction function. These measures are usually based on lexical matching and do not take into account semantic similarities between words that are lexically different. The main interest of this research is the effect of using semantics in the process of supervised text classification. This effect is evaluated through an experimental study on documents related to the medical domain using the UMLS (Unified Medical Language System) as a semantic resource. This evaluation follows four scenarios involving semantics at different steps of the classification process: the first scenario incorporates the conceptualization step where text is enriched with corresponding concepts from UMLS; both the second and the third scenarios concern enriching vectors that represent text as Bag of Concepts (BOC) with similar concepts; the last scenario considers using semantics during class prediction, where concepts as well as the relations between them are involved in decision making. We test the first scenario using three popular classification techniques: Rocchio, NB and SVM. We choose Rocchio for the other scenarios for its extendibility with semantics. According to experiment, results demonstrated significant improvement in classification performance using conceptualization before indexing. Moderate improvements are reported using conceptualized text representation with semantic enrichment after indexing or with semantic text-to-text semantic similarity measures for prediction.

Keywords.

Supervised text classification, semantics, conceptualization, semantic enrichment, semantic similarity measures, medical domain, UMLS, Rocchio, NB, SVM.

RÉSUMÉ.

Face à la multitude croissante de documents publiés sur le Web, il est apparu nécessaire de développer des techniques de classification automatique efficaces à base d'apprentissage généralement supervisé. La plupart de ces techniques de classification supervisée utilisent des sacs de mots (BOW- bags of words) en tant que modèle de représentation des textes dans l'espace vectoriel. Ce modèle comporte trois inconvénients majeurs : il considère les synonymes comme des caractéristiques distinctes, ne résout pas les ambiguïtés, et il considère les mots polysémiques comme des caractéristiques identiques. Ces inconvénients sont principalement liés à l'absence de prise en compte de la sémantique dans le modèle BOW. De plus, les mesures de similarité utilisées en tant que fonctions de prédiction par certaines techniques dans ce modèle se basent sur un appariement lexical ne tenant pas compte des similarités sémantiques entre des mots différents d'un point de vue lexical. La recherche que nous présentons ici porte sur l'impact de l'usage de la sémantique dans le processus de la classification supervisée de textes. Cet impact est évalué au travers d'une étude expérimentale sur des documents issus du domaine médical et en utilisant UMLS (Unified Medical Language System) en tant que ressource sémantique. Cette évaluation est faite selon quatre scénarii expérimentaux d'ajout de sémantique à plusieurs niveaux du processus de classification. Le premier scénario correspond à la conceptualisation où le texte est enrichi avant indexation par des concepts correspondant dans UMLS ; le deuxième et le troisième scénario concernent l'enrichissement des vecteurs représentant les textes après indexation dans un sac de concepts (BOC – bag of concepts) par des concepts similaires. Enfin le dernier scénario utilise la sémantique au niveau de la prédiction des classes, où les concepts ainsi que les relations entre eux, sont impliqués dans la prise de décision. Le premier scénario est testé en utilisant trois des méthodes de classification les plus connues : Rocchio, NB et SVM. Les trois autres scénarii sont uniquement testés en utilisant Rocchio qui est le mieux à même d'accueillir les modifications nécessaires. Au travers de ces différentes expérimentations nous avons tout d'abord montré que des améliorations significatives pouvaient être obtenues avec la conceptualisation du texte avant l'indexation. Ensuite, à partir de représentations vectorielles conceptualisées, nous avons constaté des améliorations plus modérées avec d'une part l'enrichissement sémantique de cette représentation vectorielle après indexation, et d'autre part l'usage de mesures de similarité sémantique en prédiction.

Mots clés.

La classification supervisée de texte, la sémantique, la conceptualisation, l'enrichissement sémantique, les mesures de similarité sémantique, le domaine médical, UMLS, Rocchio, NB, SVM.

REMERCIEMENTS.

Je tiens tout d'abord à exprimer ma reconnaissance à mes encadrants M. Bernard Espinasse et M. Sébastien Fournier pour avoir dirigé ce travail de recherche. Je vous remercie pour votre aide et vos conseils précieux, pour votre disponibilité et votre confiance, ainsi que pour votre gentillesse et sympathie au cours de ces années. J'ai été extrêmement sensible à vos qualités humaines d'écoute et de compréhension tout au long de ce travail doctoral.

J'exprime toute ma gratitude aux membres de jury de m'avoir honorée par leur présence. Je remercie très sincèrement Mme Sylvie Calabretto et Mme Lynda Tamine-Lechani d'avoir rapporté sur ce travail et pour leurs remarques constructives. Je remercie également Mme Nadine Cullot, M. Patrice Bellot et M. Jean-Pierre Chevallet, d'avoir accepté d'être examinateurs à la soutenance de ma thèse et d'avoir bien voulu juger ce travail.

Mes remerciements vont également à M. Moustapha Ouladsine, Directeur du LSIS, de m'avoir accueillie au sein de son laboratoire et pour ses efforts dans l'amélioration du bien-être des doctorants.

J'ai pu travailler dans un cadre particulièrement agréable, grâce à l'ensemble des membres de laboratoire LSIS, et plus particulièrement des membres de l'équipe DIMAG. Merci à tous pour votre bonne humeur et pour votre soutien moral tout au long de ma thèse. Je pense particulièrement à M. Patrice Bellot, M. Alain Ferrarini, Mme Sana Sellami pour de nombreuses discussions et pour la confiance et l'intérêt qu'ils ont manifestés à l'égard de mon travail.

Je n'oublierai pas de remercier Mme Beatrice Alcala, Mme Corine Scotto, Mme Valérie Mass et Mme Sandrine Dulac pour leur gentillesse, leur disponibilité, et pour m'avoir aidée dans les démarches administratives.

Je remercie également les membres des services techniques du laboratoire LSIS, et tout particulièrement les membres du service informatique pour leur support technique exceptionnel durant les années de ma thèse.

Mes remerciements vont également à Mme Corine Cauvet, Mme Monique Rolbert, M. Farid Nouioua et M. Eric Ronot dans le cadre de mes activités d'enseignement à l'Université d'Aix-Marseille.

Un grand merci à tous mes amis et mes collègues avec qui j'ai passé de bons moments ainsi que des périodes difficiles durant ma thèse. Merci pour vos témoignages d'amitié et pour votre soutien.

Mes dernières pensées iront vers ma famille et ma belle-famille. Merci de m'avoir accompagnée et soutenue au quotidien tout au long de ces années. Un grand merci à mes parents, qui m'ont donné le plus beau des cadeaux, sans vous et sans votre amour inconditionnel je n'en serais pas là aujourd'hui.

Enfin, Kamel, mon époux, je ne te remercierai jamais assez pour tout ce que tu as fait pour moi. Tu étais toujours là pour moi durant les bons moments ainsi que les périodes de doute pour me réconforter et m'aider à trouver des solutions. Pour tes multiples conseils et pour ton soutien affectif sans faille, pour toutes les heures que tu as consacrées à la relecture de cette thèse et pour l'espoir, le courage et la confiance que tu m'as donnés, encore merci.

Table of contents

CHAPTER 1: INTRODUCTION.....	9
1 Research context and motivation	11
2 Thesis statement	12
3 Contribution	13
4 Thesis structure.....	14
CHAPTER 2: SUPERVISED TEXT CLASSIFICATION.....	17
1 Introduction.....	19
1.1 Definitions and Foundation	19
1.2 Historical Overview	20
1.3 Chapter outline	20
2 The vector space model VSM for Text Representation.....	22
2.1 Tokenization	23
2.2 Stop words removal	24
2.3 Stemming and lemmatization	24
2.4 Weighting	24
2.5 Additional tuning	25
2.6 BOW weak points.....	25
3 Classical Supervised Text Classification Techniques	27
3.1 Rocchio	27
3.2 Support Vector Machines (SVM)	28
3.3 Naïve bayes (NB)	29
3.4 Comparison.....	30
4 Similarity Measures	32
4.1 Cosine	32
4.2 Jaccard.....	32
4.3 Pearson correlation coefficient	32
4.4 Averaged Kullback-Leibler divergence	33
4.5 Levenshtein.....	33
4.6 Conclusion	33
5 Classifier Evaluation	34
5.1 Precision, recall, F-Measure and Accuracy	34
5.2 Micro/Macro Measures	35
5.3 McNemar's Test	36
5.4 Paired Samples Student's t-test.....	36
5.5 Discussion	37
6 Testbed and Preliminary Experiments	38
6.1 Classifiers.....	38
6.2 Corpora.....	38
6.2.1 20NewsGroups corpus	38
6.2.2 Reuters	39
6.2.3 Ohsumed	40
6.3 Testing SVM, NB, and Rocchio on classical text classification corpora	40

6.3.1	Experiments on the 20NewsGroups corpus	41
6.3.2	Experiments on the Reuters corpus	43
6.3.3	Experiments on the OHSUMED corpus.....	44
6.3.4	Conclusion	45
6.4	The effect of training set labeling: case study on 20NewsGroups	46
6.4.1	Experiments on six chosen classes.....	46
6.4.2	Experiments on the corpus after reorganization	47
6.4.3	Conclusion	48
7	Conclusion	49
 CHAPTER 3: SEMANTIC TEXT CLASSIFICATION.....		51
1	Introduction.....	53
2	Semantic resources.....	55
2.1	WordNet	55
2.2	Unified Medical Language System UMLS.....	56
2.3	Wikipedia.....	58
2.4	Open Directory Program ODP (DMOZ)	59
2.5	Discussion	60
3	Semantics for text classification	62
3.1	Involving semantics in indexing	62
3.1.1	Latent topic modeling	63
3.1.2	Semantic kernels.....	64
3.1.3	Alternative features for the Vector Space Model (VSM).....	66
3.1.4	Discussion	70
3.2	Involving semantics in training	71
3.2.1	Semantic trees	72
3.2.2	Concept Forests	73
3.2.3	Discussion	73
3.3	Involving semantics in class prediction	75
3.4	Discussion	78
4	Semantic similarity measures	82
4.1	Ontology-based measures	82
4.1.1	Path-based similarity measures.....	82
4.1.2	Path and depth-based similarity measures	84
4.1.3	Discussion	86
4.2	Information theoretic measures	89
4.2.1	Computing IC-based semantic similarity measures using corpus statistics	89
4.2.2	Computing IC-based semantic similarity measures using the ontology	91
4.2.3	Discussion	92
4.3	Feature-based measures	95
4.3.1	The vision of Tversky	95
4.3.2	Feature-based semantic similarity measures	96
4.3.3	Discussion	99
4.4	Hybrid measures	101
4.4.1	Some hybrid measures	101
4.4.2	Discussion	103
4.5	Comparing families of semantic similarity measures.....	105
5	Conclusion	106

CHAPTER 4: A FRAMEWORK FOR SUPERVISED SEMANTIC TEXT CLASSIFICATION 109

1	Introduction.....	111
2	Involving semantics in supervised text classification: a conceptual framework	112
3	Involving semantics through text conceptualization.....	114
3.1	Text Conceptualization Task	114
3.1.1	Text Conceptualization Strategies	114
3.1.2	Disambiguation Strategies.....	115
3.2	Generic framework for text conceptualization	116
3.3	Conclusion	116
4	Involving semantic similarity in supervised text classification	117
4.1	Semantic similarity.....	117
4.2	Proximity matrix.....	118
4.3	Semantic kernels.....	119
4.4	Enriching vectors.....	120
4.5	Semantic measures for text-to-text similarity	123
4.6	Conclusion	125
5	Methodology	127
5.1	Scenario 1: Conceptualization only	127
5.2	Scenario 2: Conceptualization and enrichment before training	127
5.3	Scenario 3: Conceptualization and enrichment before prediction	128
5.4	Scenario 4: Conceptualization and semantic text-to-text similarity for prediction.....	129
5.5	Conclusion	129
6	Related tools in the medical domain	131
6.1	Tools for text to concept mapping	131
6.1.1	PubMed Automatic Term Mapping (ATM).....	131
6.1.2	MaxMatcher	131
6.1.3	MGREP.....	132
6.1.4	MetaMap	132
6.2	Tools for semantic similarity	134
6.2.1	Semantic similarity engine	134
6.2.2	UMLS::Similarity.....	135
6.3	Conclusion	136
7	Conclusion	138

CHAPTER 5: SEMANTIC TEXT CLASSIFICATION: EXPERIMENT IN THE MEDICAL DOMAIN 139

1	Introduction.....	141
2	Experiments applying scenario1 on Ohsumed using Rocchio, SVM and NB.....	142
2.1	Platform for supervised classification of conceptualized text.....	142
2.1.1	Text Conceptualization task	143
2.1.2	Indexing task.....	144
2.1.3	Training and classification tasks	147
2.2	Evaluating Results	147
2.2.1	Results using Rocchio with Cosine	148
2.2.2	Results using Rocchio with Jaccard	150
2.2.3	Results using Rocchio with KullbackLeibler	152

2.2.4	Results using Rocchio with Levenshtein	154
2.2.5	Results using Rocchio with Pearson	156
2.2.6	Results using NB.....	158
2.2.7	Results using SVM	160
2.2.8	Comparing MacroAveraged F1-Measure of the Classification Techniques	162
2.2.9	Comparing F1-Measure of the Classification Techniques for each class.....	164
2.2.10	Conclusion	168
3	Experiments applying scenario2 on Ohsumed using Rocchio	169
3.1	Platform for supervised text classification deploying Semantic Kernels.....	169
3.1.1	Text Conceptualization task	170
3.1.2	Proximity matrix	170
3.1.3	Enriching vectors using Semantic Kernels	172
3.2	Evaluating results.....	172
3.2.1	Observations.....	173
3.2.2	Analysis and conclusion.....	174
4	Experiments applying scenario3 on Ohsumed using Rocchio	176
4.1	Platform for supervised text classification deploying Enriching Vectors	176
4.1.1	Enriching Vectors	177
4.2	Evaluating results.....	177
4.2.1	Results using Rocchio with Cosine	177
4.2.2	Results using Rocchio with Jaccard	179
4.2.3	Results using Rocchio with Kulback	180
4.2.4	Results using Rocchio with Levenshtein.....	181
4.2.5	Results using Rocchio with Pearson	181
4.2.6	Conclusion	183
5	Experiments applying scenario4 on Ohsumed using Rocchio	185
5.1	Platform for supervised text classification deploying Semantic Text-To-Text Similarity Measures	185
5.1.1	Semantic Text-To-Text Similarity Measures	185
5.2	Evaluating results.....	186
5.2.1	Results using AvgMaxAssymldf.....	186
5.2.2	Results using AvgMaxAssymTFIDF	187
5.2.3	Conclusion	188
6	Conclusion	190
CHAPTER 6: CONCLUSION AND PERSPECTIVES		193
1	Conclusion	195
2	Contribution	197
2.1	Text conceptualization	197
2.2	Semantic enrichment before training	197
2.3	Semantic enrichment before prediction	198
2.4	Deploying semantics in prediction	198
3	Perspectives.....	199
4	List of Publications	201
REFERENCES		203

Table of figures

FIGURE 1. THE VECTOR SPACE MODEL FOR INFORMATION RETRIEVAL.....	22
FIGURE 2. STEPS FROM TEXT TO VECTOR REPRESENTATION (INDEXING), WALKING THROUGH AN EXAMPLE USING PORTER’S ALGORITHM FOR STEMMING AND TERM FREQUENCY WEIGHTING SCHEME. THE CHARACTER “ ” IS USED HERE AS A DELIMITER.	23
FIGURE 3. TEXT CLASSIFICATION: GENERAL STEPS FOR SUPERVISED TECHNIQUES	27
FIGURE 4. ROCCHIO-BASED CLASSIFICATION. C_1 : THE CENTROID OF THE CLASS 1 AND C_2 IS THE CENTROID OF CLASS 2. X IS A NEW DOCUMENT TO CLASSIFY.....	28
FIGURE 5. SUPPORT VECTOR MACHINES CLASSIFICATION ON TWO CLASSES	29
FIGURE 6. EVALUATING ROCCHIO, NB AND SVM ON 20NewsGroups CORPUS USING F1-MEASURE	41
FIGURE 7. EVALUATING ROCCHIO, NB AND SVM ON 20NewsGroups CORPUS USING PRECISION	42
FIGURE 8. EVALUATING ROCCHIO, NB AND SVM ON 20NewsGroups CORPUS USING RECALL	42
FIGURE 9. EVALUATING ROCCHIO, NB AND SVM ON REUTERS CORPUS USING F1-MEASURE	43
FIGURE 10. EVALUATING ROCCHIO, NB AND SVM ON REUTERS CORPUS USING PRECISION	43
FIGURE 11. EVALUATING ROCCHIO, NB AND SVM ON REUTERS CORPUS USING RECALL	44
FIGURE 12. EVALUATING ROCCHIO, NB AND SVM ON OHSUMED CORPUS USING F1-MEASURE	44
FIGURE 13. EVALUATING ROCCHIO, NB AND SVM ON OHSUMED CORPUS USING PRECISION	45
FIGURE 14. EVALUATING ROCCHIO, NB AND SVM ON OHSUMED CORPUS USING RECALL	45
FIGURE 15. EVALUATING FIVE SIMILARITY MEASURES ON SIX CLASSES OF 20NewsGroups (F1-MEASURE)	47
FIGURE 16. EVALUATING FIVE SIMILARITY MEASURES ON REORGANIZED 20NewsGroups (F1-MEASURE).....	47
FIGURE 17. PART OF WORDNET WITH HYPERNYMY AND HYPONYMY RELATIONS.	56
FIGURE 18. THE VARIOUS RESOURCES AND SUBDOMAINS UNIFIED IN UMLS	57
FIGURE 19. WIKIPEDIA: PAGE FOR “CLASSIFICATION” WITH LINKS TO DIFFERENT ARTICLES RELATED TO DIFFERENT LANGUAGES, DOMAINS AND CONTEXTS OF USAGE.....	58
FIGURE 20. ODP HOME PAGE. GENERAL CONCEPTS ARE IN BOLD (2013).	60
FIGURE 21. INVOLVING SEMANTIC RESOURCES IN SUPERVISED TEXT CLASSIFICATION SYSTEM: A GENERAL ARCHITECTURE	62
FIGURE 22. MAPPING WORDS THAT OCCURRED IN TEXT TO THEIR CORRESPONDING SYNSETS IN WORDNET AND ACCUMULATING THEIR WEIGHTS WHEN MULTIPLE WORDS ARE MAPPED TO THE SAME SYNSET LIKE GOVERNMENT AND POLITICS. THEN, ACCUMULATED WEIGHTS ARE NORMALIZED AND PROPAGATED ON THE HIERARCHY (PENG <i>ET AL.</i> , 2005)	72
FIGURE 23. BUILDING A CONCEPT FOREST FOR A TEXT DOCUMENT THAT CONTAINS THE WORDS: “INFLUENZA”, “DISEASE”, “SICKNESS”, “DRUG”, “MEDICINE” (J. Z. WANG <i>ET AL.</i> , 2007).	73
FIGURE 24. A PART OF UMLS (PEDERSEN <i>ET AL.</i> , 2012). THE CONCEPT “BACTERIAL INFECTION” IS THE MOST SPECIFIC COMMON ABSTRACTION (MSCA) OF “TETANUS” AND “STREP THROAT”.....	83
FIGURE 25. A PART OF UMLS IC OF EACH CONCEPT IS CALCULATED USING A MEDICAL CORPUS ACCORDING TO (RESNIK, 1995; PEDERSEN <i>ET AL.</i> , 2012)	90
FIGURE 26. COMMON CHARACTERISTICS AMONG TWO CONCEPTS	96
FIGURE 27. SETS OF COMMON AND DISTINCTIVE CHARACTERISTICS OF CONCEPTS C_1 , C_2	96
FIGURE 28 A CONCEPTUAL FRAMEWORK TO INTEGRATE SEMANTICS IN SUPERVISED TEXT CLASSIFICATION PROCESS.	113
FIGURE 29. GENERIC PLATFORM FOR TEXT CONCEPTUALIZATION	116
FIGURE 30. BUILDING PROXIMITY MATRIX FOR A VOCABULARY OF CONCEPTS OF SIZE N.....	118
FIGURE 31. APPLYING SEMANTIC KERNEL TO A DOCUMENT VECTOR	119
FIGURE 32. STEPS TO APPLY SEMANTIC KERNEL TO A CONCEPTUALIZED TEXT DOCUMENT	120
FIGURE 33. APPLYING ENRICHING VECTORS TO A PAIR OF DOCUMENTS. AS A RESULT, THE WEIGHT CORRESPONDING TO C2 IN A CHANGES FROM 0 TO wa2 AND THE WEIGHT CORRESPONDING TO C3 IN B CHANGES FROM 0 TO wb3 . THE VOCABULARY SIZE IS LIMITED TO 4.....	121
FIGURE 34. STEPS TO APPLY ENRICHING VECTORS TO A PAIR OF CONCEPTUALIZED TEXT DOCUMENTS.....	123
FIGURE 35. STEPS TO APPLYING AGGREGATION FUNCTION ON A PAIR OF CONCEPTUALIZED DOCUMENTS.	123
FIGURE 36. GENERIC FRAMEWORK FOR USING TEXT CONCEPTUALIZATION IN SUPERVISED TEXT CLASSIFICATION.....	127
FIGURE 37. GENERIC FRAMEWORK USING SEMANTIC KERNELS TO ENRICH TEXT REPRESENTATION	128
FIGURE 38. GENERIC FRAMEWORK USING ENRICHING VECTORS TO ENRICH TEXT REPRESENTATION	128
FIGURE 39. GENERIC FRAMEWORK FOR USING SEMANTIC TEXT-TO-TEXT SIMILARITY IN CLASS PREDICTION	129
FIGURE 40. CONCEPT PROCESSING IN MGREP (DAI, 2008)	132

FIGURE 41. METAMap: STEPS FOR TEXT TO CONCEPT MAPPING (ARONSON <i>ET AL.</i> , 2010). THE EXAMPLE OF COMMAND LINE OUTPUT OF METAMap OCCURRED USING THE PHRASE “PATIENTS WITH HEARING LOSS”	133
FIGURE 42. SEMANTIC SIMILARITY ENGINE WITH A CACHE DATABASE FOR BUILDING PROXIMITY MATRIX	135
FIGURE 43. ACTIVITY DIAGRAM OF THE SEMANTIC SIMILARITY ENGINE	135
FIGURE 44. COMPONENTS INSIDE THE SEMANTIC SIMILARITY ENGINE FOR THE MEDICAL DOMAIN	136
FIGURE 45. THE ARCHITECTURE OF A PLATFORM FOR CONCEPTUALIZED TEXT CLASSIFICATION.	142
FIGURE 46. 12 STRATEGIES FOR TEXT CONCEPTUALIZATION USING METAMap: A WALK THROUGH AN EXAMPLE. FOR THE UTTERANCE “WITH HEARING LOSS” WE CHOSE TO USE A MAXIMUM OF TWO MAPPINGS TO AVOID CONFUSION.....	143
FIGURE 47. CONCEPTUALIZATION: THE PROCESS STEP BY STEP	144
FIGURE 48. INDEXING PROCESS: STEP BY STEP	144
FIGURE 49. EVALUATING THE EFFECT OF VOCABULARY SIZE THAT VARIES FROM [100 TO 4000] FEATURES ON CLASSIFICATION RESULTS (F1-MEASURE) USING ROCCHIO WITH COSINE ON OHSUMED TEXTUAL CORPUS	146
FIGURE 50. EVALUATING THE EFFECT OF VOCABULARY SIZE THAT VARIES FROM [100 TO 4000] FEATURES ON CLASSIFICATION RESULTS (F1-MEASURE) USING ROCCHIO WITH COSINE ON OHSUMED CONCEPTUALIZED CORPUS ACCORDING TO THE STRATEGY (“COMPLETE”, “BEST”, “IDS”).....	146
FIGURE 51. NUMBER OF CLASSES WITH IMPROVED F1-MEASURE ON CONCEPTUALIZED TEXT COMPARED WITH THE ORIGINAL TEXT USING ROCCHIO WITH COSINE SIMILARITY MEASURE	149
FIGURE 52. NUMBER OF CLASSES WITH IMPROVED F1-MEASURE ON CONCEPTUALIZED TEXT COMPARED WITH THE ORIGINAL TEXT USING ROCCHIO WITH JACCARD SIMILARITY MEASURE	152
FIGURE 53. NUMBER OF CLASSES WITH IMPROVED F1-MEASURE ON CONCEPTUALIZED TEXT COMPARED WITH THE ORIGINAL TEXT USING ROCCHIO WITH KULLBACKLEIBLER SIMILARITY MEASURE	154
FIGURE 54. NUMBER OF CLASSES WITH IMPROVED F1-MEASURE ON CONCEPTUALIZED TEXT COMPARED WITH THE ORIGINAL TEXT USING ROCCHIO WITH LEVENSHTAIN SIMILARITY MEASURE	156
FIGURE 55. NUMBER OF CLASSES WITH IMPROVED F1-MEASURE ON CONCEPTUALIZED TEXT COMPARED WITH THE ORIGINAL TEXT USING ROCCHIO WITH PEARSON SIMILARITY MEASURE	157
FIGURE 56. NUMBER OF CLASSES WITH IMPROVED F1-MEASURE ON CONCEPTUALIZED TEXT COMPARED WITH THE ORIGINAL TEXT USING NB.....	159
FIGURE 57. NUMBER OF CLASSES WITH IMPROVED F1-MEASURE ON CONCEPTUALIZED TEXT COMPARED WITH THE ORIGINAL TEXT USING SVM.....	162
FIGURE 58. PERCENTAGE OF SHARE OF EACH CLASSIFICATION TECHNIQUE ON THE TOTAL NUMBER OF CASES WHERE AN INCREASE IN F1-MEASURE OCCURRED. CASES ARE GATHERED FROM FORMER SECTIONS.....	164
FIGURE 59. THE NUMBER OF CASES WHERE AN INCREASE IN F1-MEASURE OCCURRED FOR EACH CLASS AFTER TESTING CLASSIFIERS ON ALL CONCEPTUALIZED VERSIONS OF OHSUMED.	165
FIGURE 60. PLATFORM FOR SUPERVISED TEXT CLASSIFICATION DEPLOYING SEMANTIC KERNELS	169
FIGURE 61. RESULTS OF APPLYING SEMANTIC KERNELS USING CDIST, LCH, NAM, WUP, ZHONG SEMANTIC SIMILARITY MEASURES AND FIVE VARIANTS OF ROCCHIO.....	173
FIGURE 62. PLATFORM FOR SUPERVISED TEXT CLASSIFICATION DEPLOYING ENRICHING VECTORS.....	176
FIGURE 63. NUMBER OF IMPROVED CLASSES AFTER APPLYING ENRICHING VECTORS ON ROCCHIO WITH COSINE USING FIVE SEMANTIC SIMILARITY MEASURES	179
FIGURE 64. NUMBER OF IMPROVED CLASSES AFTER APPLYING ENRICHING VECTORS ON ROCCHIO WITH JACCARD USING FIVE SEMANTIC SIMILARITY MEASURES	180
FIGURE 65. NUMBER OF IMPROVED CLASSES AFTER APPLYING ENRICHING VECTORS ON ROCCHIO WITH PEARSON USING FIVE SEMANTIC SIMILARITY MEASURES	183
FIGURE 66. PLATFORM FOR SUPERVISED TEXT CLASSIFICATION DEPLOYING SEMANTIC SIMILARITY MEASURES	185
FIGURE 67. NUMBER OF IMPROVED CLASSES AFTER APPLYING ROCCHIO WITH <i>AVGMAXASYMTFIDF</i> FOR PREDICTION	188

Table of tables

TABLE 1. COMPARING THREE CLASSIFICATION TECHNIQUES	31
TABLE 2. CONFUSION MATRIX COMPOSITION	34
TABLE 3. CONTINGENCY TABLE OF TWO CLASSIFIERS A, B	36
TABLE 4. CONTINGENCY TABLE OF TWO CLASSIFIERS A, B UNDER THE NULL HYPOTHESIS	36
TABLE 5. TWENTY ACTUALITY CLASSES OF 20NEWSGROUPS CORPUS	39
TABLE 6. REUTERS-21578 CORPUS	40
TABLE 7. OHSUMED CORPUS	40
TABLE 8. COMPARING FOUR SEMANTIC RESOURCES: WORDNET, UMLS, WIKIPEDIA AND ODP.	60
TABLE 9. TWO DOCUMENTS (d1 , d2) TERM VECTORS. NUMBERS ARE TERM FREQUENCIES IN DOCUMENT ..	65
TABLE 10. SEMANTIC SIMILARITY MATRIX FOR THREE TERMS: PUMA, COUGAR, FELINE.	65
TABLE 11. TWO DOCUMENTS (d1 , d2) TERM VECTORS. NUMBERS REPRESENT WEIGHTS AFTER INNER PRODUCT BETWEEN A LINE FROM TABLE 9 AND A COLUMN FROM TABLE 10	66
TABLE 12. COMPARING ALTERNATIVE FEATURES OF THE VSM. (+, ++, +++): DEGREES OF SUPPORT (-): UNSUPPORTED CRITERION	70
TABLE 13. COMPARING LATENT TOPIC MODELING, SEMANTIC KERNELS AND ALTERNATIVE FEATURES FOR INTEGRATION SEMANTICS IN TEXT INDEXING	71
TABLE 14. COMPARING GENERALIZATION, ENRICHING VECTORS, SEMANTIC TREES AND CONCEPT FORESTS IN INVOLVING SEMANTICS IN TRAINING	74
TABLE 15 INVOLVING SEMANTICS IN TEXT REPRESENTATION COMPARISON AND IN LEARNING CLASS MODEL	81
TABLE 16. STRUCTURE-BASED SIMILARITY MEASURES	88
TABLE 17. IC-BASED SIMILARITY MEASURES	94
TABLE 18. DIFFERENT SCENARIOS OF TVERSKY SIMILARITY MEASURE	97
TABLE 19. XML DESCRIPTIONS OF “HYPOTHYROIDISM” AND “HYPERTHYROIDISM” FROM WORDNET AND MESH (PETRAKIS ET AL., 2006)	98
TABLE 20. FEATURE-BASED SIMILARITY MEASURES	100
TABLE 21. MAPPING BETWEEN FEATURE-BASED AND IC SIMILARITY MODELS (PIRRO ET AL., 2010)	101
TABLE 22. MAPPING BETWEEN SET-BASED SIMILARITY COEFFICIENTS AND IC-BASED COEFFICIENTS	102
TABLE 23. HYBRID SIMILARITY MEASURES	104
TABLE 24. COMPARISON BETWEEN STRUCTURE, IC, AND FEATURE-BASED SIMILARITY MEASURES	105
TABLE 25. COMPARING FOUR TOOLS FOR TEXT TO UMLS CONCEPT MAPPING	137
TABLE 26. TRANSFORM THE PHRASE “PATIENTS WITH HEARING LOSS” INTO WORD/FREQUENCY VECTOR BEFORE AND AFTER CONCEPTUALIZATION USING THE 12 CONCEPTUALIZATION STRATEGIES.	145
TABLE 27. RESULTS OF APPLYING ROCCHIO WITH COSINE SIMILARITY MEASURE TO OHSUMED CORPUS AND TO THE RESULTS OF ITS CONCEPTUALIZATION ACCORDING TO 12 CONCEPTUALIZATION STRATEGIES. (*) DENOTES SIGNIFICANCE ACCORDING TO McNEMAR TEST. VALUES IN THE TABLE ARE PERCENTAGES.	148
TABLE 28. RESULTS OF APPLYING ROCCHIO WITH JACCARD SIMILARITY MEASURE TO OHSUMED CORPUS AND TO THE RESULTS OF ITS CONCEPTUALIZATION ACCORDING TO 12 CONCEPTUALIZATION STRATEGIES. (*) DENOTES SIGNIFICANCE ACCORDING TO McNEMAR TEST. VALUES IN THE TABLE ARE PERCENTAGES.	150
TABLE 29. RESULTS OF APPLYING ROCCHIO WITH KULLBACKLEIBLER SIMILARITY MEASURE TO OHSUMED CORPUS AND TO THE RESULTS OF ITS CONCEPTUALIZATION ACCORDING TO 12 CONCEPTUALIZATION STRATEGIES. (*) DENOTES SIGNIFICANCE ACCORDING TO McNEMAR. VALUES IN THE TABLE ARE PERCENTAGES.	153
TABLE 30. RESULTS OF APPLYING ROCCHIO WITH LEVENSHTAIN SIMILARITY MEASURE TO OHSUMED CORPUS AND TO THE RESULTS OF ITS CONCEPTUALIZATION ACCORDING TO 12 CONCEPTUALIZATION STRATEGIES. (*) DENOTES SIGNIFICANCE ACCORDING TO McNEMAR TEST. VALUES IN THE TABLE ARE PERCENTAGES.	155
TABLE 31. RESULTS OF APPLYING ROCCHIO WITH PEARSON SIMILARITY MEASURE TO OHSUMED CORPUS AND TO THE RESULTS OF ITS CONCEPTUALIZATION ACCORDING TO 12 CONCEPTUALIZATION STRATEGIES. (*) DENOTES SIGNIFICANCE ACCORDING TO McNEMAR TEST. VALUES IN THE TABLE ARE PERCENTAGES.	156
TABLE 32. RESULTS OF APPLYING NB TO OHSUMED CORPUS AND TO THE RESULTS OF ITS CONCEPTUALIZATION ACCORDING TO 12 CONCEPTUALIZATION STRATEGIES. (*) DENOTES SIGNIFICANCE ACCORDING TO McNEMAR TEST. VALUES IN THE TABLE ARE PERCENTAGES.	158
TABLE 33. RESULTS OF APPLYING SVM TO OHSUMED CORPUS AND TO THE RESULTS OF ITS CONCEPTUALIZATION ACCORDING TO 12 CONCEPTUALIZATION STRATEGIES. (*) DENOTES SIGNIFICANCE ACCORDING TO McNEMAR. VALUES IN THE TABLE ARE PERCENTAGES.	161

TABLE 34. MACROAVERAGED F1-MEASURE FOR 7 CLASSIFICATION TECHNIQUES APPLIED TO THE ORIGINAL OHSUMED CORPUS AND TO THE RESULTS OF ITS CONCEPTUALIZATION ACCORDING TO 12 CONCEPTUALIZATION STRATEGIES. (*) DENOTES SIGNIFICANCE ACCORDING TO T-TEST (YANG ET AL., 1999). VALUES IN THE TABLE ARE PERCENTAGES.	163
TABLE 35. F1-MEASURE VALUES FOR EACH CLASS USING 7 DIFFERENT CLASSIFIERS AND 12 CONCEPTUALIZATION STRATEGIES. (*) DENOTES THAT CLASSIFIER'S PERFORMANCE ON THE CONCEPTUALIZED OHSUMED IS SIGNIFICANTLY DIFFERENT FROM ITS PERFORMANCE ON THE ORIGINAL OHSUMED ACCORDING TO McNEMAR TEST WITH A EQUALS TO (0.05). INCREASED F1-MEASURE IS IN BOLD WITH A LIGHT RED BACKGROUND.	167
TABLE 36. FIVE SEMANTIC SIMILARITY MEASURES: INTERVALS AND OBSERVATIONS ON THEIR VALUES ..	170
TABLE 37. A SUBSET OF 30 MEDICAL CONCEPT PAIRS MANUALLY RATED BY MEDICAL EXPERTS AND PHYSICIANS FOR SEMANTIC SIMILARITY	171
TABLE 38. SPEARMAN'S CORRELATION BETWEEN FIVE SIMILARITY MEASURES AND HUMAN JUDGMENT ON PEDERSEN'S CORPUS (PEDERSEN ET AL., 2012).	172
TABLE 39. RESULTS OF APPLYING ROCCHIO WITH COSINE SIMILARITY MEASURE TO OHSUMED CORPUS AND TO THE RESULTS OF ITS COMPLETE CONCEPTUALIZATION WITH ENRICHING VECTORS. (*) DENOTES SIGNIFICANCE ACCORDING TO McNEMAR TEST. VALUES IN THE TABLE ARE PERCENTAGES.	178
TABLE 40. RESULTS OF APPLYING ROCCHIO WITH JACCARD SIMILARITY MEASURE TO OHSUMED CORPUS AND TO THE RESULTS OF ITS COMPLETE CONCEPTUALIZATION WITH ENRICHING VECTORS. (*) DENOTES SIGNIFICANCE ACCORDING TO McNEMAR TEST. VALUES IN THE TABLE ARE PERCENTAGES.	179
TABLE 41. RESULTS OF APPLYING ROCCHIO WITH KULLBACKLEIBLER SIMILARITY MEASURE TO OHSUMED CORPUS AND TO THE RESULTS OF ITS COMPLETE CONCEPTUALIZATION WITH ENRICHING VECTORS. (*) DENOTES SIGNIFICANCE ACCORDING TO McNEMAR TEST. VALUES IN THE TABLE ARE PERCENTAGES.	181
TABLE 42. RESULTS OF APPLYING ROCCHIO WITH LEVENSHTAIN SIMILARITY MEASURE TO OHSUMED CORPUS AND TO THE RESULTS OF ITS COMPLETE CONCEPTUALIZATION WITH ENRICHING VECTORS. (*) DENOTES SIGNIFICANCE ACCORDING TO McNEMAR TEST. VALUES IN THE TABLE ARE PERCENTAGES.	181
TABLE 43. RESULTS OF APPLYING ROCCHIO WITH PEARSON SIMILARITY MEASURE TO OHSUMED CORPUS AND TO THE RESULTS OF ITS COMPLETE CONCEPTUALIZATION WITH ENRICHING VECTORS. (*) DENOTES SIGNIFICANCE ACCORDING TO McNEMAR TEST. VALUES IN THE TABLE ARE PERCENTAGES.	182
TABLE 44. RESULTS OF APPLYING ROCCHIO WITH AVGMAXASSYMIIDF SEMANTIC SIMILARITY MEASURE TO OHSUMED CORPUS AND TO THE RESULTS OF ITS COMPLETE CONCEPTUALIZATION. (*) DENOTES SIGNIFICANCE ACCORDING TO McNEMAR TEST. VALUES IN THE TABLE ARE PERCENTAGES.	187
TABLE 45. RESULTS OF APPLYING ROCCHIO WITH AVGMAXASSYMTFIDF SEMANTIC SIMILARITY MEASURE TO OHSUMED CORPUS AND TO THE RESULTS OF ITS COMPLETE CONCEPTUALIZATION. (*) DENOTES SIGNIFICANCE ACCORDING TO McNEMAR TEST. VALUES IN THE TABLE ARE PERCENTAGES.	187

CHAPTER 1: INTRODUCTION

1 Research context and motivation

The notion of Classification dates back to the work of Plato, who proposed to classify objects according to their common characteristics. Throughout the past centuries, the notion of classification and categorization gained great interest, and especially thematic text classification, as people realized its importance in facilitating information access and interpretation, even for a small number of documents. Computers and information technologies improved our capability to accumulate and store information since the work of Plato, which makes text classification and organization into meaningful topics an effort demanding and time-consuming task. Moreover, the increasing availability of electronic documents and the rapid growth of the web made document automatic classification a key method for organizing information and knowledge discovery in order to meet our increasing capacity to collect them.

During the last century, Rule-based expert systems replaced manual classification; this limited the role of domain experts to the process of writing these rules. Nevertheless, rule implementation and maintenance is a labor intensive and a time consuming task (Manning *et al.*, 2008) which led to supervised text classification techniques that require a sample of categorized documents, known by a training corpus, to learn the classification rules or the classification model. Thus, many techniques for *supervised classification* appeared aiming to classify and organize text documents into classes using their characteristics imitating domain experts.

Usually, text is represented in the vector space as bag of words (BOW) (G. Salton *et al.*, 1975) by the words it mentions, each being weighted according to how often it occurs in the text. Their positions and order of occurrences are not considered. This model has been the most popular way to represent textual content for Information Retrieval (IR), Clustering and supervised Classification. In the BOW, texts are considered similar if they share enough characteristics (or words).

As compared with human perception of information, BOW has two drawbacks (L. Huang *et al.*, 2012). The first drawback is *ambiguity*; it pays no attention to the fact that different words may have the same sense while the same word may have different senses according to its context. Humans can straightforwardly resolve ambiguities and interpret the conveyed meaning of such words using the knowledge obtained from previous experience. Second, the model is *orthogonal*: it ignores relations between words and treats them independently. In fact, words are always related to each other to form a meaningful idea which facilitates our understanding of text.

This thesis investigates *semantic approaches* for overcoming drawbacks of the BOW model by replacing words with concepts as features describing text contents, in the aim to improve text classification effectiveness. Concepts are explicit units of knowledge that constitute along with the explicit relations between them a controlled vocabulary or a semantic resource that can be either general purpose or domain specific. *Concepts* are unambiguous and relations between them are explicitly defined and can be quantified, this makes concepts the best alternative feature for the VSM (Bloehdorn *et al.*, 2006; L. Huang *et al.*, 2012).

We call techniques that use concepts and their relations to improve classification semantic text classification, to distinguish them from the traditional word-based models. This

thesis investigates how semantic resources can be deployed to improve text classification, and how they enrich the classification process to take semantic relations as well as concepts into account.

2 Thesis statement

This thesis claims that:

Using concepts in text representation and taking the relations among them into account during the classification process can significantly improve the effectiveness of text classification, using classical classification techniques.

Demonstrating evidence to support this claim involves two parts: first, use concepts to represent texts instead/with words in the VSM; and second, take their relations into account in the classification process. This thesis treated these parts in four different steps or scenarios:

First, semantic knowledge is involved in indexing through **Conceptualization**: the process of finding a match or a relevant concept in a semantic resource that conveys the meaning of a word or multiple words from text. This process resolves ambiguities in text and identifies matched concepts that convey the accurate meaning. Different strategies might be appropriate for **Conceptualization** and **Disambiguation** (Bloehdorn *et al.*, 2006) involving semantics in text representation in different manners. Keeping only concepts in text transforms the classical BOW to a Bag of Concepts (BOC) where concepts are the only descriptors of text.

Second scenario involves the semantic relations between concepts in enriching text representation in the VSM as a BOC. This scenario aims to investigate the impact of enriching text representation by means of **Semantic Kernels** (Wang *et al.*, 2008) that can be applied on vectors representing the training corpus and the test documents after indexing. After involving similar concepts from the semantic resource in text representation, training and classification phases are executed to assess the influence of this enrichment on text classification effectiveness.

Third scenario is quite similar to the second one except for the fact that enrichment is done just before prediction and can be used with classification techniques having a vector-like classification model. Thus, it applies the approach **Enriching Vectors** (L. Huang *et al.*, 2012) in order to mutually enrich two BOCs with similar concepts from the semantic resource. After involving similar concepts from the semantic resource in text representation and in the model, classes for new documents are predicted and compared with the results that were obtained using the original BOC. This scenario aims to assess the influence of this enrichment on text classification effectiveness.

Forth, this thesis investigates the effectiveness of **Semantic Measures for Text-To-Text Similarity** (Mihalcea *et al.*, 2006) instead of classical similarity measures that are usually used in prediction for classification in the VSM. These measures use semantic similarities among concepts -that are assessed utilizing the relations between them- instead of lexical matching of classical similarity measures that ignore relations between features of the representation model. This scenario aims to assess the influence of using **Semantic Measures for Text-To-Text Similarity** on text classification effectiveness in the VSM.

Despite the great interest in semantic text classification, integrating semantics in classification is a subject of debate as works in the literature seem to disagree on its utility (Stein *et al.*, 2006). Nevertheless, it seems to be promising to take the application domain into consideration when developing a system for semantic classification (Ferretti *et al.*, 2008) for two reasons: first, many researchers faced difficulties in classifying domain specific text documents (Bloehdorn *et al.*, 2006; Bai *et al.*, 2010). Second, many researchers reported that using domain specific semantic resources improves classification effectiveness (Bloehdorn *et al.*, 2006; Aseervatham *et al.*, 2009; Guisse *et al.*, 2009). Thus, this thesis investigates the effect of involving semantics in text classification applied in the medical domain.

We employ three standard datasets that are widely used for evaluating classification techniques in our preliminary experiments (see chapter 2): Reuters collection, 20Newsgroup collection and Ohsumed collection of medical abstracts. In the three collections, the classes of documents are related to their textual contents or in other words are thematic classes. The preliminary experiments discuss challenges in supervised text classification and propose solutions aiming at more effective text classification.

As for experiments in the medical domain involving semantics, we use Ohsumed collection of medical abstracts (Hersh *et al.*, 1994) and the Unified Medical Language System (UMLS®) (2013) as the semantic resource. We use statistical measures for evaluating classification results and the significance of improvement in classification effectiveness after applying the four preceding scenarios. This evaluation provides a guide for the application of our approaches in practice.

The process of text classification in the VSM produces three major artifacts: text representation, classification model, and similarity for class prediction. This thesis aims to involve semantics, including concepts and relations among them- in the first and the last artifact. Thus, the classification model is the only artifact that is not considered explicitly in this work, yet it is influenced by the semantics used in text representation. For other classification techniques evaluated in this work, semantics are involved in text representation only for reasons of extendibility.

3 Contribution

In general, text classification is tackled using syntactic and statistical information only ignoring semantics that reside in text and keeps problems like redundancy and ambiguities unresolved. Text classification is a challenging task in a sparse and high dimensional feature space.

In this thesis, we aim to investigate where and how to involve semantics in order to facilitate text classification and to what extent it can help in better classification. Through the previously presented scenarios, this thesis studies the following points:

- First, semantic resources may be useful at text indexing step so index would contain words, concepts or a combination of both forms. This thesis investigates these issues through conceptualization step that is applied to plain text before indexing. Different strategies for text conceptualization resulted in different text representation; this may have influences on classification effectiveness. This study concludes with

recommendations on the use of concepts in text representations for three classical techniques SVM, NB and Rocchio.

- Second, concepts are not independent; they are interrelated in the semantic resources by different types of relations. These relations connect similar concepts that can contribute to more effective text classification if involved in the classification process. This point investigates semantic enrichment of text representation using similar concept and its influence on classification effectiveness. This work applies *Semantic Kernels* that is usually used with SVM (Wang *et al.*, 2008) to Rocchio and applies *Enriching Vectors* that was tested on KNN and K-Means to Rocchio.
- Third, semantic relations can also be beneficial in class prediction. In fact, an aggregation of semantic similarities between concepts representing two vectors can be used as a semantic text-to-text similarity measure in the vector space and can be used in Rocchio's prediction. Classical similarity measures, like Cosine, depend on the common features between the compared texts only and treat features independently which makes semantic similarity measures more adequate to compare BOCs. This work applies a state of the art *Semantic Text-To-Text Similarity Measures* and a new semantic measure on Rocchio and investigated the influence of such measures on the effectiveness of Rocchio. This part concludes with recommendations on the use of aggregation function on semantic similarities between concepts as a prediction criterion using BOC model.

4 Thesis structure

This thesis is structured in four main chapters: *Supervised Text Classification* (Chapter 2): an experimental study on popular classification techniques and collections to identify challenges in text classification, *Semantic Text Classification* (Chapter 3): an overview of the state of the art approaches involving semantics in text classification, *A Framework for Supervised Semantic Text Classification* (Chapter 4) our methodology for involving semantics in the classification process, and *Semantic Text Classification: Experiment In The Medical Domain* (Chapter 5): experimental study applying our methodology in the medical domain and evaluates the influence of semantics on classification effectiveness. The details of this structure are as follows:

Chapter 2 Supervised Text Classification presents an experimental study on three classical classification techniques on three different corpora in order to identify challenges in supervised text classification. Section 1 presents some definitions of the notion of classification from its origins to its modern foundations and particularly in the context of automatic text classification. Section 2 presents the vector space model, a traditional model for text representation. Section 3 presents and compares three classical classification techniques Rocchio, NB and SVM. Section 4 introduces five popular similarity measures that assess the similarity between two vectors in the vector space model which is a prediction criterion of some classification techniques in the VSM. Section 5 presents some measures for evaluating classification effectiveness and statistical tests of significance. Section 6 concerns technical details of the testbed we deployed and the experiments on the three classification techniques presented in section 3. Finally, this chapter concludes with a discussion and conclusions on

preliminary results identifying the limits of classical text classification and proposing solutions to overcome them.

Chapter 3 Semantic Text Classification presents an overview of the state of the art works involving semantics in text classification. Section 2 presents some semantic resources already used in semantic text classification in some details. Section 3 presents different state of art approaches involving semantic knowledge in text classification and similar tasks related to IR. These approaches deploy semantic resources at different steps in the process of text classification: text representation, training and in classification as well. Section 4 presents a state of the art on semantic similarity measures that assess the semantic similarity between pairs of concepts in the semantic resource. This semantic similarity is deployed in many state of the art approaches presented in section 3 in order to involve semantics in text classification.

Chapter 4 A Framework for Supervised Semantic Text Classification is the conceptual contribution of this thesis on the use of semantics in text classification. This chapter presents our methodology towards a semantic text classification. Section 2 presents a conceptual framework for involving semantics (concepts and relations among them) in text classification at different steps of its process. Section 3 presents specifications for involving semantics in text representation through conceptualization and disambiguation. Section 4 focuses on deploying semantic similarity measures in addition to concepts in text classification through representation enrichment and semantic text-to-text similarity, all using proximity matrix. Section 5 presents the methodology with which we intend to carry out the experimental study in next chapter. Here, we identify four different scenarios. Section 6 presents different tools for text to concept mapping in the medical domain and UMLS::Similarity module for computing semantic similarities on UMLS. These tools are essential to implement scenarios in corresponding platforms in order to carry out the experiments and test the different approaches in the medical domain.

Chapter 5 Semantic Text Classification: Experiment In The Medical Domain presents our experimental study that applies the methodology presented in chapter 4 in four different scenarios. section 2 presents experiments on Ohsumed after conceptualization in a platform implementing the first scenario and using three different classification techniques. Section 3 presents experiments on Ohsumed using Semantic Kernels for enrichment and Rocchio for classification; this section applies the second scenario. Section 4 presents experiments on Ohsumed using Enriching Vectors for enrichment and Rocchio for classification and implementing the third scenario. Section 5 presents experiments on Ohsumed using semantic similarity measures for class prediction implementing the fourth scenario on previous chapter. This chapter concludes with a discussion on the influence of semantics on text classification.

In conclusion, we present a summary on the research that was done in this thesis presenting our major scientific contribution in the domain of semantic text classification. Finally, we present the possible future works through short, medium and long term prospects.

CHAPTER 2: SUPERVISED TEXT CLASSIFICATION

Table of contents

1	Introduction.....	19
1.1	Definitions and Foundation	19
1.2	Historical Overview	20
1.3	Chapter outline	20
2	The vector space model VSM for Text Representation.....	22
2.1	Tokenization	23
2.2	Stop words removal	24
2.3	Stemming and lemmatization	24
2.4	Weighting	24
2.5	Additional tuning	25
2.6	BOW weak points.....	25
3	Classical Supervised Text Classification Techniques	27
3.1	Rocchio	27
3.2	Support Vector Machines (SVM).....	28
3.3	Naïve bayes (NB).....	29
3.4	Comparison.....	30
4	Similarity Measures	32
4.1	Cosine.....	32
4.2	Jaccard.....	32
4.3	Pearson correlation coefficient	32
4.4	Averaged Kullback-Leibler divergence	33
4.5	Levenshtein.....	33
4.6	Conclusion	33
5	Classifier Evaluation	34
5.1	Precision, recall, F-Measure and Accuracy	34
5.2	Micro/Macro Measures	35
5.3	McNemar's Test	36
5.4	Paired Samples Student's t-test.....	36
5.5	Discussion	37
6	Testbed and Preliminary Experiments	38
6.1	Classifiers.....	38
6.2	Corpora.....	38
6.2.1	20NewsGroups corpus	38
6.2.2	Reuters	39
6.2.3	Ohsumed	40
6.3	Testing SVM, NB, and Rocchio on classical text classification corpora	40
6.3.1	Experiments on the 20NewsGroups corpus	41
6.3.2	Experiments on the Reuters corpus	43
6.3.3	Experiments on the OHSUMED corpus.....	44
6.3.4	Conclusion	45
6.4	The effect of training set labeling: case study on 20NewsGroups	46
6.4.1	Experiments on six chosen classes.....	46
6.4.2	Experiments on the corpus after reorganization	47
6.4.3	Conclusion	48

1 Introduction

Text document classification is vital for organizing and archiving information since the ancient civilizations. Nowadays, many researchers are interested in developing approaches for efficient automatic text classification especially with the exploding increase in electronic text documents on the internet. This section introduces the notion of classification through state of the arte definitions and presents a historical overview on the development of document classification from a manual task to an automatic and efficient one thanks to computers. Finally, this section presents an outline for the rest of this chapter.

1.1 Definitions and Foundation

The notion of *Classification* appeared for the first time in the work of Plato, who proposed a classification approach for organizing objects according to their similar properties. Aristotle in his “Categories” treatise (Aristotle) explored and developed this notion; he analyzes in details the common and the distinctive features of objects defining from a logical point of view different categories and classes. Aristotle also applied this definition on his studies in biology to classify living beings. Some of his classes are still in use today.

Throughout the centuries, the notion of classification and categorization gained great interest and led to multiple theories and hypothesis. Both terms have many definitions; some of them are similar, complementary and sometimes conflicting. Authors in (Manning *et al.*, 2008) define classification as follows: “*Given a set of classes, we seek to determine which class(es) a given object belongs to.*”.

According to (Borko *et al.*, 1963): “*The problem of automatic document classification is a part of the larger problem of automatic content analysis. Classification means the determination of subject content. For a document to be classified under a given heading, it must be ascertained that its subject matter relates to that area of discourse. In most cases this is a relatively easy decision for a human being to make. The question being raised is whether a computer can be programmed to determine the subject content of a document and the category (categories) into which it should be classified.*”.

In the context of *Information Retrieval* (IR), the notion of *Text Classification* has also many definitions in the literature. According to *Sebastiani* (Sebastiani, 2005) “*Text categorization (also known as text classification or topic spotting) is the task of automatically sorting a set of documents into categories from a predefined set.*”. *Sebastiani* gave also another definition in (Sebastiani, 2002) “*The automated categorization (or classification) of texts into predefined categories.*”. In the literature, authors use different terms to refer to the same notion and the same definition like *text categorization*, *topic classification*, or *topic spotting*.

In this work, we choose to use “*Text Classification*” to refer to content-based classification of text documents; given a text document and a set of predetermined classes, text classification searches the most appropriate class to this document according to its contents. Text classification is a vital task in the IR domain as it is central to different tasks like email filtering, sentiment analysis, topic-specific search, information extraction and so forth (Manning *et al.*, 2008; Albitar *et al.*, 2010; Espinasse *et al.*, 2011).

1.2 Historical Overview

Before computer, classification tasks have been solved manually by experts. A librarian organizes library books and documents by assigning them specific categories or notations based on the classification system in use in his library (Dewey, 2011). Thanks to the digital revolution, an alternative approach based on rules helped in classification (Prabowo *et al.*, 2002; Taghva *et al.*, 2003). Indeed, rule-based expert systems have good scaling properties as compared to manual classification. These systems are based on handcrafted rules for classification made by experts. Generally, classification rules relate the occurrence of certain keywords or "features" in a document to a specific class. However, rule implementation and maintenance demand a lot of time and effort from domain experts, in addition to their limited adaptability to any changes in their domain and for each new domain of application (Pierre, 2001; Manning *et al.*, 2008).

Consequently, learning-based techniques appeared, introducing new methods for classification also known by machine learning techniques or statistical techniques. In the literature, two families of these techniques can be distinguished: supervised and unsupervised techniques.

Unsupervised techniques can discover classes or categories in a collection of text documents. Some techniques need a prior knowledge on the number of classes to discover like K-means (MacQueen, 1967) while others make no prior assumptions like ISODATA (Ball *et al.*, 1965). Members of this family are known by Clustering techniques (Manning *et al.*, 2008).

Supervised techniques use training sets to learn decision models that can discriminate relevant classes. The teacher to these techniques is the domain expert that labels each document with one of the predetermined set of classes. The classes and the set of labeled documents are required by this family of classifiers and are considered as a priori knowledge. These models are often crystallized in induced rules, or statistical estimations. Such supervised methods require training set preparation through manual labeling, that associates its documents to their relevant classes. Even if this preparation effort is significant, it is nevertheless less effort and time demanding if compared with rule implementation by domain experts (Manning *et al.*, 2008).

In this study, we were interested in supervised techniques for text classification. Many works propose new techniques or ameliorations applied to classical ones like Rocchio, SVM, NB, Decision trees, artificial neural network, genetic algorithm and so forth (Baharudin *et al.*, 2010). Due to their popularity, we will mainly focus on the first three techniques in the rest of this work.

1.3 Chapter outline

So far, this chapter presented some definitions of the notion of classification from its origins to its modern foundations and particularly in the context of automatic text classification. Next section presents the vector space model, a well-known model for text representation and that is used by the three classical text classification techniques presented and compared in third section. Section four introduces five popular similarity measures that assess the similarity

between two vectors in the vector space model which is essential to text classification using all of the three classical techniques. Section five presents some statistics for evaluating classification effectiveness. Section six concerns technical details of the testbed we deployed and the experiments on the three classifiers. We finish this chapter with a discussion and conclusions on preliminary results identifying the limits of these classifiers and proposing solutions to overcome them.

2 The vector space model VSM for Text Representation

Most supervised classification techniques use the Vector Space Model (VSM) (G. Salton *et al.*, 1975) to represent text documents. According to David Dubin, Gerard Salton's publication on VSM is "*The most influential paper Gerard Salton never wrote*" (Dubin, 2004). The SMART system proposed by Salton was a revolutionary progress for information retrieval. In his book "*Automatic Text Processing*" (Gerard Salton, 1989), Salton defines the process of information retrieval through the following points:

- Queries and documents are represented in a VSM by vectors, each of them composed of a set of terms.
- The term elements composing a vector are assigned a weight that can be either binary (1 for the presence and 0 for the absence of the term) or a number implying the importance of the term in the represented text.
- Similarity is computed in order to assess the relevance of a document to a particular query.

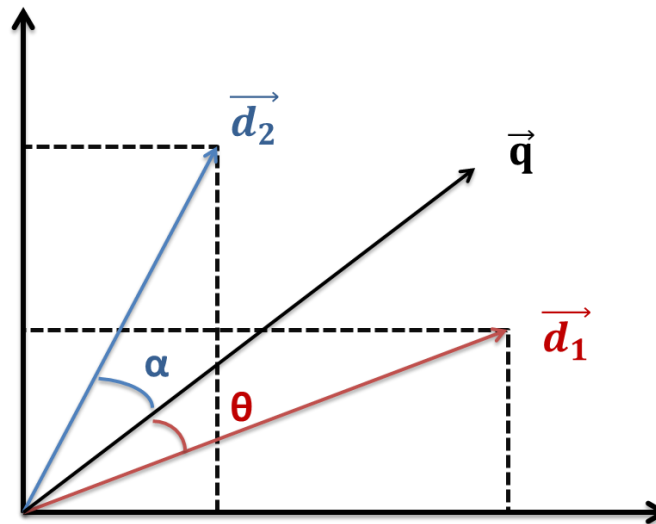


Figure 1. The Vector Space Model for Information Retrieval

Using Cosine (G. Salton *et al.*, 1975) for example as a similarity measure, the relevance of a document to a query is estimated by the cosine of the angle between the vectors that represent them in the VSM. Its relevance is assessed using the dot product of these vectors. Given two documents $\vec{d}_1, \vec{d}_2$ and a query, $\vec{q}$, $\vec{d}_1$ can be considered more relevant than $\vec{d}_2$ if $(\cos \theta > \cos \alpha)$. This example is illustrated in Figure 1.

So the components of vectors describe the textual data and the similarity measures like cosine or other computations describe how the resulting IR system works so the vector space model can provide a very general and flexible abstraction for such systems (Dubin, 2004).

Besides his experimentations on VSM in the IR domain, Salton also investigated its utility in other areas (Dubin, 2004) like book indexing, clustering, automatic linking and

relevance feedback and many other areas. As for relevance feedback, the experimentations on the VSM were realized by J.J. Rocchio (G. Salton, 1971). The proposed model which was named after him as Rocchio was adapted later to text classification and known by Centroid-based Classification which is of a great interest in this work.

Plain text to vector transformation, which is known by indexing as well, passes through multiple steps: tokenization, stop word removal, stemming and weighting in order to get the final vector or index that represents the initial text in the vector space. The following subsections present these steps in details. A walk through an example is illustrated in Figure 2 as well. Each text document is represented by a sparse high dimensional vector; each dimension corresponds to a particular word or other type of features like phrases or concepts. Features of the first systems using this model were principally words, and vectors of the VSM are so considered as Bags of Words (BOW).

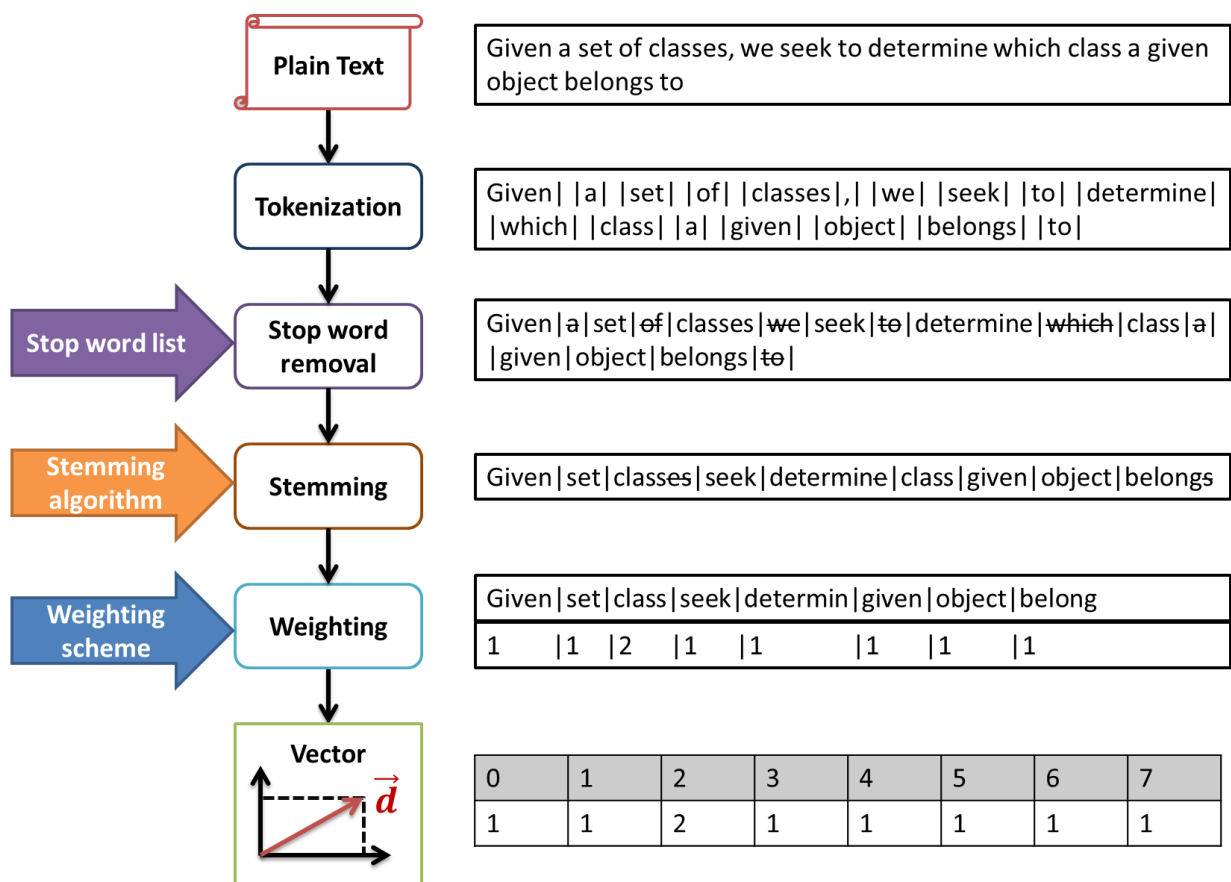


Figure 2. Steps from text to vector representation (indexing), walking through an example using porter's algorithm for stemming and term frequency weighting scheme. The character "|" is used here as a delimiter.

2.1 Tokenization

Tokenization, by definition, is the task of chopping up plain text into character sequences called *tokens*. In general, tokenization chops on whitespaces and throws away some characters like punctuation (Manning *et al.*, 2008; Baharudin *et al.*, 2010). Similar tokens are called *types* and

at the end of vector creation the normalized types are transformed into *terms* that constitute the BOW's vocabulary.

Tokenizers have to deal with many linguistic issues like language identification, which character to chop on (apostrophe, hyphens, etc.) and also deal with special information like dates, names of places and others where whitespaces and special characters are non-separating (Manning *et al.*, 2008). An example of tokenization is illustrated in the first step of indexing (see Figure 2).

2.2 Stop words removal

After tokenization, many common words appear to be not very useful for text document representation as they are considered semantically non selective (like a, an, and, etc.). These words are called stop words and are eliminated from the vocabulary in this step. Lists of stop words vary in length according to the context from long lists (300 words) to relatively short ones (20 words). On the contrary, web search engines don't remove any stop word as they can be used in web page ranking (Manning *et al.*, 2008).

2.3 Stemming and lemmatization

Many tokens retrieved from previous steps can be derivations of the same word like the verb *classify* and the noun *class* and also the inflections of the verb *like* and its past tense *liked*. These different forms are related to lexical and grammatical reasons respectively, and usually it is useful to be considered the same in indexing. In order to reduce these inflectional or derivational forms of words, either Stemming or Lemmatization can be used.

Stemming is a heuristic algorithm that removes inflectional affixes from words by chopping off their endings. A well-known algorithm is *Porter Stemmer* for English (Porter, 1980). Lemmatization uses usually a dictionary and a NLP morphological analyzer to this end. Both methods have the same goal: put similar words in their common base form. Nevertheless their results differ: Lemmatization results are real words whereas Stemming might result in character sequences with no meaning.

2.4 Weighting

Former steps result in a set of terms that constitute the model's vocabulary. These terms are considered as the dimensions of the VSM. From this point of view, each document can be represented by a vector where each of its components reflects the importance of the corresponding term in the document. In the literature, many weighting schemes were used varying from binary representation indicating the presence or the absence of a term in the document to normalized statistical weighted schemes. Here we cite some of these schemes (Lan *et al.*, 2009) like *tf*, *idf*, *idf-prob*, *Odds Ratio*, χ^2 etc.

The most popular weighting scheme is *tf.idf* (Gerard Salton, 1989). The basic hypothesis of this scheme is that the term frequency may not be sufficient for discriminating relevant documents from others (Lan *et al.*, 2009). To overcome this limitation, the term frequency is multiplied by the factor Inverse Document Frequency *idf*. In fact, this factor varies

inversely with the number of documents that contain a particular term so it can improve the discriminative power of the term frequency. Given the term t_j in document d_i *tf.idf* score is estimated as follows:

$$w_{ij} = \text{tf}_{ij} * \log(N/\text{df}_j) \quad (1)$$

tf_{ij} : Frequency of term t_j in document d_i .

N: Number of documents.

df_j : Number of documents that contain term t_j .

In the context of supervised text classification, training set is usually used to estimate this factor so df_j is the number of the documents that contain the term and are labeled as relevant to a particular class in the training set and N is the number of documents labeled as relevant to the same class.

The result of applying vector space modeling to a text document is a weighted vector of features:

$$d_i = (w_{i1}, w_{i2}, w_{i3}, \dots, w_{in}) \quad (2)$$

2.5 Additional tuning

To equally evaluate terms occurring in two documents with different lengths, normalization is vital to the weighting scheme. Term frequency can be divided by the document length so the occurrence of a term is judged frequent relatively to the sum of frequencies of all the other terms constituting the document. In fact, normalization can attenuate some weights that may be biased.

In addition to weights, feature selection or dimensionality reduction techniques make classifiers focus more on important features and ignore noisy ones that don't contribute to decision making and may sometimes decrease classification accuracy (Yang *et al.*, 1997; Guyon *et al.*, 2003; Geng *et al.*, 2007). The number of dimensions of the VSM can also affect the efficiency of the classifier and slows down decision making. A good feature selection method should take into consideration the classification technique as well as the application domain (Baharudin *et al.*, 2010).

2.6 BOW weak points

BOW is the most commonly used text representation in almost every field that involves text analysis like IR, classification, clustering, etc. However, this model has some well-known limitations (Bloehdorn *et al.*, 2006; L. Huang *et al.*, 2012):

- Synonymy: also called term mismatch problem or *redundancy* problem. In general, different texts use different words to express the same concept. Since the BOW does not connect synonyms, these words are considered different terms.
- Polysemy: also called semantic *ambiguity*. In all languages, a word can have different meanings depending on its surrounding context. Since, the BOW does not capture such differences. So the same word with two different meanings is considered a single term.
- Relations between words: The BOW model ignores the connections between words: it assumes that they are independent of each other. This problem is known by

orthogonality. The relations cover the synonymy, hyponymy and polysemy relations among other senses of relatedness between words.

These three limitations can affect not only the representation accuracy and the similarities among documents but also the robustness of the model. For example, if a new document shares no term with the used vocabulary so it wouldn't be properly classified. Many works proposed solutions to overcome these limitations. This will be discussed later in chapter 3.

3 Classical Supervised Text Classification Techniques

In general, supervised classification techniques need to learn a classification model for each context in order to classify new documents in the same context. To learn the classification model, a collection of documents representing the context is labeled with the appropriate classes according to their contents by a domain expert. Then, this collection, known by training set, helps the techniques learn and generalize a model based on documents labels and contents. These steps constitute the *training phase*. During the *test phase*, also known by the *classification phase*, a new document is presented to the classifier that, depending on document contents and the learned model, predicts the document's class. In both phases, text is transformed into vectors through *Indexing* step. These phases are illustrated in Figure 3

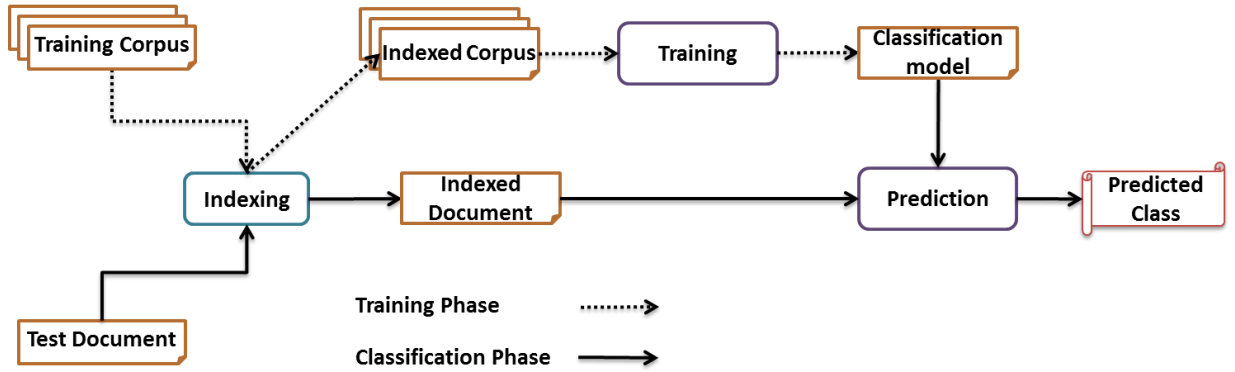


Figure 3. Text classification: General steps for supervised techniques

This section presents in details three classical text classification techniques: Rocchio, SVM and NB all using the vector space model for text representation. Finally, we present a comparative study on these techniques.

3.1 Rocchio

Rocchio or centroid based classification (Han *et al.*, 2000) for text documents is widely used in Information Retrieval tasks, in particular for relevance feedback where it was investigated for the first time by *J.J.Rocchio* (G. Salton, 1971). Afterwards it was adapted for text classification.

For centroid-based classification, each class is represented by a vector positioned at the center of the sphere delimited by training documents related to this class. This vector is so called the class's centroid as it summarized all features of the class as collected during learning phase, through vectors representing training documents, following the BOW as detailed earlier. Having n classes in the training corpus, n centroid vectors $\{C_1, C_2, \dots, C_n\}$ are calculated through the training phase by means of the following formula (Sebastiani, 2002):

$$w_{ki} = \beta \cdot \sum_{d_j \in \text{POS}_i} \frac{w_{kj}}{\|\text{POS}_i\|} - \gamma \cdot \sum_{d_j \in \text{NEG}_i} \frac{w_{kj}}{\|\text{NEG}_i\|} \quad (3)$$

w_{ki} : the weight of term t_k in the centroid of the class C_i

w_{kj} : the weight of term t_k in document d_j

POS_i, NEG_i : positive and negative examples of class c_i

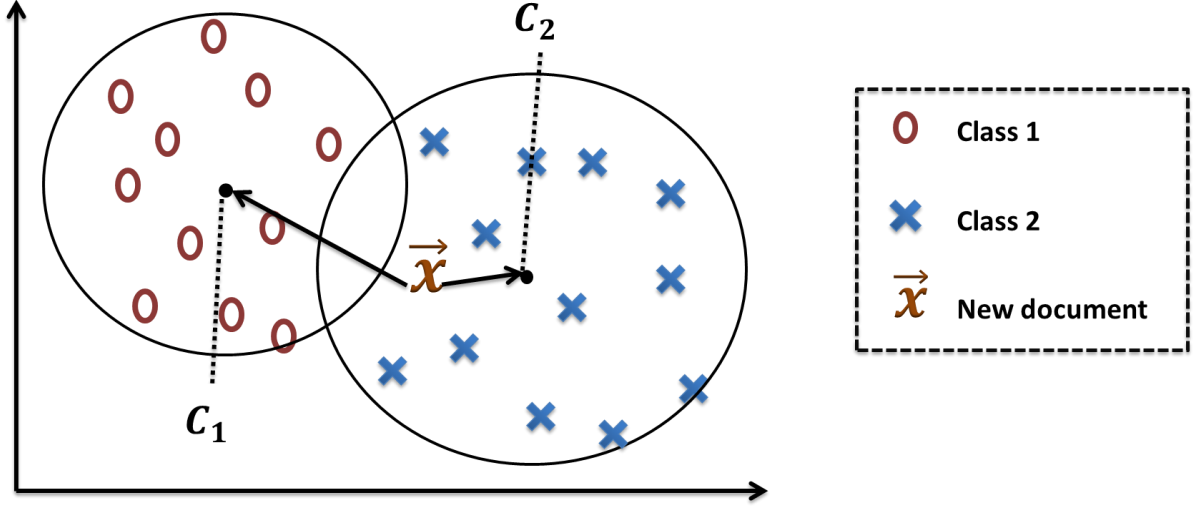


Figure 4. Rocchio-based classification. C_1 : the centroid of the class 1 and C_2 is the centroid of class 2. x is a new document to classify

In this work we use the following parameters ($\beta = 1, \gamma = 0$) focusing particularly on positive examples ($\beta = 1$) (Han *et al.*, 2000; Sebastiani, 2002).

In order to classify a new document x , first we use the TF/IDF weighting scheme to calculate the vector representing this document in the space. Then, resulting vector is compared to all centroids of n candidate classes using a similarity measure (see section 4). So the class of the document x is the one represented by the most similar centroid; its centroid ($\vec{c}_i$) maximizes the similarity function ($SimFun$) with the vector of the document (see equation (4))

$$C = \arg \max_{i=1,2,\dots,n} (SimFun(\vec{x}, \vec{c}_i)) \quad (4)$$

As illustrated in Figure 4, the centroid C_2 is more similar to the new document x than C_1 (closer according to the Euclidian distance) so Rocchio assigns *Class 2* to x .

3.2 Support Vector Machines (SVM)

The *Support Vector Machines* (SVM) (V. N. Vapnik, 1995; Burges; V. Vapnik, 1998) is a supervised technique that tries to find the borderline between two classes using the vectors of their documents as represented in the VSM. In cases where these classes are linearly separable, the SVM seek a hyperplane that determines the borderline between them and that maximizes the margins, or in other words the maximal separation between classes, so the resulting classifier is called *maximum margin classifier*. Maximal margins help minimize the classification error risk. Samples at the margins are the support vectors after which the technique was named. Given two classes of examples (X and O) that are linearly separable, the hyperplane that separates the examples ($\vec{w}\vec{x} - b = 0$) represents the classification model as illustrated in Figure 5. SVM are naturally two-class classifiers. Nevertheless, many works adapted them to multiclass classifiers using a set of one-versus-all classifiers (Duan *et al.*, 2005).

The number of training examples and the number of features affect the efficiency of SVM. This is a great concern in text classification where text is usually represented in a high dimensional feature space. In order to limit the computation load, it is necessary to eliminate noisy examples and features from the training set (Manning *et al.*, 2008). Furthermore, some training sets are not linearly separable by SVM. Thus, it is common to use the kernel trick to simplify the task and to project the training set into a higher dimensional space where the classifier can find a linear solution (Manning *et al.*, 2008). Since SVM uses the dot product of example vectors in the original space ($\vec{w}\vec{x} - b = 0$), a kernel function corresponds to a dot product in some expanded feature space. We mention the popular *radial basis function* (RBF) that we use later in our experiments (see equation (5)) (Chang *et al.*, 2011).

$$K(\vec{x}_i, \vec{x}_j) = \exp(-\gamma \|\vec{x}_i - \vec{x}_j\|^2), \quad \gamma > 0 \quad (5)$$

γ is a parameter, $\vec{x}_i, \vec{x}_j$ two examples in the original space

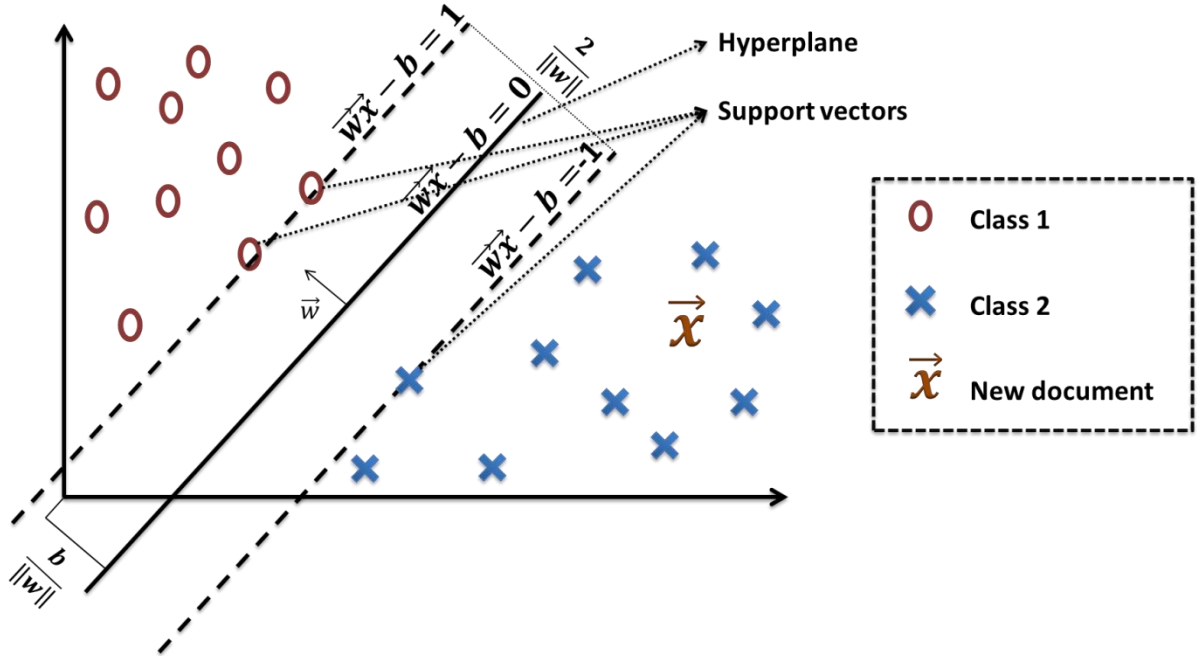


Figure 5. Support vector machines classification on two classes

3.3 Naïve bayes (NB)

Naïve Bayes (NB) classification (Lewis, 1998) is a supervised probabilistic technique for classification. The decision criterion of this technique is the probability that a document belongs to a particular class. This probability is given in the following equation:

$$P(c|d) = P(c) \prod_{1 \leq k \leq n_d} P(t_k|c) \quad (6)$$

Where c is a class and d is a document.

$P(t_k|c)$ is the conditional probability that the term t_k , that occurs in the document d , occurs in the class c or in other words it estimates the relevance of t_k to a particular class.

Depending on the training set with N documents, the preceding probabilities are calculated as follows:

$$\hat{P}(c) = N_c / N \quad (7)$$

$$\hat{P}(t_k|c) = (T_{ct} + 1) / \sum_{t' \in V} (T_{ct'} + 1) \quad (8)$$

Where N_c is the number of documents having the label c in the training set. T_{ct} is the frequency of term t_k in the documents labeled by c . V is the vocabulary of terms found in the training set. $\hat{P}(c)$ is the estimated value of $P(c)$.

Using a training set, NB learns a probabilistic model on class distribution. For every new document, this technique represents it by a binary vector reflecting the presence and the absence of vocabulary terms (1,0 respectively) in the documents. Applying the learned model, NB calculates the probabilities that the new document belongs to each of the possible classes. Finally, NB assigns to the new document the class with the maximum probability.

3.4 Comparison

To compare the preceding three classification techniques, we retain this set of characteristics that we use in Table 1 as criterion of comparison:

- Complexity: the complexity of the classifier's algorithm
- Representation: text representation model
- Basic hypothesis: the information needed by the classification technique to build a classification model or to classify
- Decision making: how to choose the appropriate class
- Decision criterion: the criterion used in choosing the appropriate class
- The effect of training set characteristics: on training or classification in terms of execution time
- Effect of noisy examples: the influence of such examples on the classification technique

Despite *NB*'s (Lewis, 1998) attractive simplicity and efficiency, this classifier, also called "The Binary Independence Model", has many critical weaknesses. First of all, the unrealistic independence hypothesis of this model considers each feature independently for calculating their occurrence probabilities related to a class. Second, binary vectors used for document representation neglect information that can be derived from terms' frequencies in the processed document or even its length (Lewis, 1998). Thus, many works propose different variations of this model to overcome its limitations (Sebastiani, 2002).

As for text classification using *SVM* (V. N. Vapnik, 1995; Burges, 1998; V. Vapnik, 1998), the number of features characterizing documents is crucial to learning efficiency as it can significantly increment its complexity. So it is essential to this method to eliminate noisy and irrelevant features that might have negative influence on complexity and also on classification results (Manning *et al.*, 2008). Consequently, SVM is considered a time and memory consuming method for text classification where class discrimination needs a considerable set of features (Manning *et al.*, 2008). However, SVM is a very effective and largely used technique for classification.

Criteria\Technique	NB	Rocchio	SVM
Complexity	Simple	Average	Complex
Representation	Binary vectors	BOW	BOW
Basic hypothesis	-Probabilistic model -Parametric	-Vector distribution in the space -Direct test	-Vector distribution in the space - Supervised learning
Decision making	The most probable class	The class with the most similar centroid	The class residing at one side of the hyperplane
Decision criterion	Conditional probability	Similarity Measure like Cosine	The position of the document's vector
The effect of training set characteristics	Small training set is sufficient	training documents distribution determines class boundaries	Large training set → slows down training
Effect of noisy examples	Insignificant	Insignificant	Insignificant

Table 1. Comparing three classification techniques.

Compared to other methods for text classification, Rocchio (or centroid-based classifier) has many advantages (Han *et al.*, 2000). First, learned classification model summarizes the characteristics of each class through a centroid vector, even if these characteristics aren't all present simultaneously in all documents. This summarization is relatively absent in other classification methods except for NB that learns term-probability distribution functions summing up their occurrences in different classes. Another advantage is the use of similarity measure that compares a document to class centroids taking into account summarization result as well as term occurrences in the document in order to classify it. NB uses learned probability distribution only to estimate the occurrence probability of each term independently to other terms in a class summarization or to document co-occurring terms. Nevertheless, the basic assumption of Rocchio on the regularity in class distribution is considered its major drawback; in some cases, the centroids it learns depending on training documents might be insufficient for classification.

In next section, we test SVM and NB and Rocchio (using different similarity measures) on three corpora: 20NewsGroups, Reuters and Ohsumed. We will compare their performance in different contexts and identify their strengths and weaknesses empirically. Our objective in this work is to propose solutions to improve their performance depending on the conclusions of this chapter.

4 Similarity Measures

Many document classification and document clustering techniques deploy similarity measures to estimate the similarity between a document and a class prototype (A. Huang, 2008). In the VSM, these measures assess the similarity between a document vector and the vector representing a class or its centroid. The following subsections introduce five popular similarity measures (Cosine, Jaccard, Pearson, Kullback Leibler, and Levenshtein) that we deploy later in section 6 in experiments with Rocchio.

4.1 Cosine

Cosine is the most popular similarity measure and largely used in information retrieval, document clustering, and document classification research domains.

Having two vectors $A(a_1, a_2, \dots, a_n)$, $B(b_1, b_2, \dots, b_n)$, the similarity between these vector is estimated using the cosine of the angle (t) that they delimit:

$$\text{Sim}_{\text{Cosine}}(A, B) = \cos(t) = \frac{A \cdot B}{|A| * |B|} \quad (9)$$

Where:

$$A \cdot B = \sum_i a_i * b_i \quad |A| = \sqrt{\sum_i a_i^2}$$

$i \in [0, n-1]$; n : the number of features in vector space.

In systems using this similarity measure, changing documents' length has no influence on the result as the angle they delimit is still the same.

4.2 Jaccard

Jaccard estimates the similarity to the division of the intersection by the union. According to ensemble theory, given two ensembles (X, Y) the similarity between them is estimated according to the following equation:

$$\text{Jaccard}(X, Y) = \frac{X \cap Y}{X \cup Y} \quad (10)$$

Having two vectors $A(a_1, a_2, \dots, a_n)$, $B(b_1, b_2, \dots, b_n)$, according to Jaccard the similarity between A and B is by definition:

$$\text{Sim}_{\text{Jaccard}}(A, B) = \frac{A \cdot B}{|A|^2 + |B|^2 - A \cdot B} \quad (11)$$

Where:

$$A \cdot B = \sum_i a_i * b_i \quad |A|^2 = \sum_i a_i^2$$

$i \in [0, n-1]$; n : the number of features in the vector space.

4.3 Pearson correlation coefficient

Given two vectors $A(a_1, a_2, \dots, a_n)$, $B(b_1, b_2, \dots, b_n)$, Pearson calculates the correlation between these vectors. Deriving their centric vectors: $A(a_1 - \bar{a}, \dots, a_n - \bar{a})$ and $B(b_1 - \bar{b}, \dots, b_n - \bar{b})$

Where: $\bar{a}$ is the average of all A's features, $\bar{b}$ is the average of all B's features.

Pearson correlation coefficient is by definition the cosine of the angle α between the centric vectors as follows:

$$\text{Sim}_{\text{Pearson}} = \frac{n \sum a_i b_i - \sum a_i \sum b_i}{\sqrt{[n \sum a_i^2 - (\sum a_i)^2][n \sum b_i^2 - (\sum b_i)^2]}} \quad (12)$$

4.4 Averaged Kullback-Leibler divergence

According to probability and information theory, Kullback-Leibler divergence is a measure estimating dis-similarities between two probability distributions. In the particular case of text processing, this measure calculates the divergence between feature distributions in documents. Given vectors' representations of their features distribution $A(a_1, a_2, \dots, a_n)$, $B(b_1, b_2, \dots, b_n)$, the divergence is calculated as follows

$$\text{Sim}_{\text{KullbackLeibler}}(A, B) = \sum_i (\pi_1 * D(a_i || w_i) + \pi_2 * D(b_i || w_i)) \quad (13)$$

Where:

$\pi_1 = \frac{a_i}{a_i + b_i}$	$\pi_2 = \frac{b_i}{a_i + b_i}$
$w_i = \pi_1 * a_i + \pi_2 * b_i$	$D(a_i w_i) = a_i * \log\left(\frac{a_i}{w_i}\right)$

4.5 Levenshtein

Levenshtein is usually used to compare two strings. A possible extension for vector comparison can be derived using the following equation given two vectors $A(a_1, a_2, \dots, a_n)$, $B(b_1, b_2, \dots, b_n)$:

$$\text{Sim}_{\text{Levenshtein}}(A, B) = 1 - (\text{Distance}/\text{Max}) \quad (14)$$

Where:

$\text{Distance}(A, B) = \sum a_i - b_i $	$\text{Max}(A, B) = \sum \text{Max}(a_i, b_i)$
--	--

4.6 Conclusion

This section presented five different similarity measures that are usually used in the literature to compare vectors in the VSM. Rocchio is one of the classification techniques that use these measures. We will test Rocchio in our experiments using each of the preceding similarity measures. In other words, we will use five different variants of Rocchio in our experiments.

5 Classifier Evaluation

During training phase, classification techniques learn classifiers or classification models that can be applied to new documents presented to it in test phase. At the end of test, the performance of the used classifier is evaluated according to its results. Evaluation involves statistical measures that enable comparing classifiers. Here we present the state of the art of commonly used evaluation measures for text classification tasks.

5.1 Precision, recall, F-Measure and Accuracy

Considering a particular class of test documents (the documents of other classes are considered as negative examples) we obtain different statistics on results: the number of correctly recognized class documents (true positives t_p), the number of correctly recognized documents that do not belong to the class (true negatives t_n), and documents that either were incorrectly assigned to the class (false positives f_p) or that were not recognized as class documents (false negatives f_n). The former four counts are the base of our evaluation measures Precision, Recall, F_β -Measure and accuracy. Table 2 illustrates what is called a confusion matrix that is composed of these four counts as well.

Class documents	Classified as Positive	Classified as Negative
Positive examples	t_p	f_n
Negative examples	f_p	t_n

Table 2. Confusion matrix composition

In this work we adopted four evaluation measures: *Precision*, *Recall*, *Accuracy* and *F_β-Measure*. In fact, *F_β-Measure* is a weighted harmonic mean of *Precision* and *Recall* and is usually used with ($\beta = 1$). These measures can be calculated as follows:

$$\text{Precision} = \frac{t_p}{t_p + f_p} \quad (15)$$

$$\text{Recall} = \frac{t_p}{t_p + f_n} \quad (16)$$

$$F_\beta - \text{Measure} = \frac{(\beta^2 + 1) * \text{Precision} * \text{Recall}}{\beta * \text{Precision} + \text{Recall}} \quad (17)$$

$$\text{Accuracy} = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \quad (18)$$

Having two classes to distinguish, effectiveness is usually measured by accuracy that measures the percentage of correct classification decisions. However, in case of more than two classes, it is more adequate to use the other measures like precision, recall and F1-Measure that give a better interpretation of the classification results (Sebastiani, 2002).

5.2 Micro/Macro Measures

In text classification with a set of different categories $C = \{c_1, c_2, c_3, \dots, c_n\}$, classifier effectiveness is evaluated using Precision, Recall or F1-Measure for each category. Evaluation results must be averaged across different categories. We refer to the sets of true positives, true negatives, false positives and false negative examples for the category c_i using TP_i, TN_i, FP_i, FN_i respectively.

In Microaveraging, categories participate in the average proportionally to the number of their positive examples (Sebastiani, 2002, 2005). This applies to both *MicroAvgPrecision* and *MicroAvgRecall* (equations (19), (20) respectively).

$$\text{MicroAvgPrecision} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + FP_i} \quad (19)$$

$$\text{MicroAvgRecall} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + FN_i} \quad (20)$$

On the contrary, for *Macroaveraging* all categories count the same; frequent and infrequent categories participate equally in *MacroAvgPrecision* and *MacroAvgRecall* (equations (21), (22) respectively) (Sebastiani, 2002, 2005). $\text{Precision}_i, \text{Recall}_i$ are related to the category c_i .

$$\text{MacroAvgPrecision} = \frac{\sum_{i=1}^{|C|} \text{Precision}_i}{|C|} = \frac{\sum_{i=1}^{|C|} \frac{TP_i}{TP_i + FP_i}}{|C|} \quad (21)$$

$$\text{MacroAvgRecall} = \frac{\sum_{i=1}^{|C|} \text{Recall}_i}{|C|} = \frac{\sum_{i=1}^{|C|} \frac{TP_i}{TP_i + FN_i}}{|C|} \quad (22)$$

Most classification techniques emphasize on either Precision or Recall, therefore we use their combination in the F_β -Measure which is more significant. Usually, researches use F_1 -Measure as the harmonic mean of precision and recall. *MicroAvgF $_\beta$ -Measure* and *MacroAvgF $_\beta$ -Measure* are calculated according to equations

$$\text{MicroAvgF}_\beta\text{Measure} = \frac{(\beta^2+1) * \text{MicroAvgPrecision} * \text{MicroAvgRecall}}{\beta^2 * \text{MicroAvgPrecision} + \text{MicroAvgRecall}} \quad (23)$$

$$\text{MacroAvgF}_\beta\text{Measure} = \frac{(\beta^2+1) * \text{MacroAvgPrecision} * \text{MacroAvgRecall}}{\beta^2 * \text{MacroAvgPrecision} + \text{MacroAvgRecall}} \quad (24)$$

In fact, Microaveraging favors classifiers with good behavior on categories that are heavily populated with document while Macroaveraging favors those with good behavior on poorly populated categories. In general, developing classifiers that behave well on poorly populated categories is very challenging therefore most research use Macroaveraging for evaluation (Sebastiani, 2002, 2005).

Given two trained classifiers on the same training set that are tested on the same test set giving Macroaveraged F1-Measure of 78 and 76 percent, respectively, to claim that the first classifier is significantly better than the second we need statistical evidence to support it. Thus, we present two statistical tests: McNemar and T-test to compare the performance of classifiers pair-to-pair.

5.3 McNemar's Test

McNemar's test (Everitt, 1992; Dietterich, 1998) is a simple way to test marginal homogeneity in K*K tables which implies that row totals are equal to the corresponding column totals.

This test is widely applied in comparing classifiers as it applies to contingency tables (Dietterich, 1998). Having two classifiers A, B trained on the same training set, when we test them on the same test set we record their results for each example in the following contingency table:

N_{00} number of examples misclassified by both A&B	N_{01} number of examples misclassified by A but not by B
N_{10} number of examples misclassified by B but not by A	N_{11} number of examples misclassified by neither A nor B

Table 3. Contingency table of two classifiers A, B.

Where the size of the test set: $N = N_{00} + N_{01} + N_{10} + N_{11}$.

Under the marginal homogeneity, both classifiers should have the same error rate leading to $N_{01} = N_{10}$ which means that the expected counts under the null hypothesis H_0 where both classifiers have the same error rate are the following:

N_{00}	$(N_{01} + N_{10})/2$
$(N_{01} + N_{10})/2$	N_{11}

Table 4. Contingency table of two classifiers A, B under the null hypothesis

In fact, McNemar test is based on a Chi-Square χ^2 that compares the distribution of the expected counts to the observed ones with a 1 degree of freedom according to the following equation:

$$\chi^2 = \frac{(|N_{01} - N_{10}| - 1)^2}{N_{01} + N_{10}} \quad (25)$$

The level of significance (α) is the probability of rejecting the null hypothesis when it is true. The tabulated value for Chi-Square with 1 degree of freedom and a level of significance $\alpha = 0.05$ is $\chi^2_{1,0.95} = 3.841459$. The confidence interval is: $0.95 = 1 - 0.05$.

The simplest way to do this test is to compare the calculated value of χ^2 with the tabulated one $\chi^2_{1,0.95}$ and if the calculated $\chi^2 \geq \chi^2_{1,0.95}$, we may reject the null hypothesis in favor of the alternative. In the context of this thesis the null hypothesis is that the compared classifiers aren't different whereas the alternative hypothesis is that the tested classifiers have a significantly different performance even when trained using the same training set. The level of significance (Type I error) of a statistical test is the probability of rejecting the null hypothesis H_0 when it is true. We will use the level of significance ($\alpha = 0.05$) in forthcoming tests which is the commonly accepted value of error in the literature (Yang *et al.*, 1999).

5.4 Paired Samples Student's t-test

This test is the most popular in machine learning literature (Dietterich, 1998; Yang *et al.*, 1999). It is used to compare two dependent samples; that is, when there are two samples that have been

"paired" or when two measures are tested on the same sample. Comparing two classifiers by of their detailed results on all categories, the compared values are collected from the documents of the same category which enables us to choose the paired samples t-test.

In fact, this test compares all pairs and calculates their differences that are then used to produce the t value as the following:

$$t = \frac{\overline{X_D}}{s_D / \sqrt{n}} \quad (26)$$

Where n is the sample size. $n - 1$ is the degree of freedom

$\overline{X_D}$ is the average

s_D is the standard deviation

According to the value of t , we can reject the null hypothesis (the compared classifiers are similar) in favor of the alternative if $|t| > t_{n-1, \alpha}$; if the calculated t is greater than the critical value of t or the tabulated t we conclude that the compared classifiers are significantly different in the evaluated context. Similarly to the preceding test, we will use the level of significance ($\alpha = 0.05$) in forthcoming tests.

5.5 Discussion

This section introduced the notion of Micro/Macro averaging that is widely used for comparing classifiers as they aggregate the by-category results into one value. In addition, this section introduced two statistical test McNemar and paired samples student's t-test that are usually used to evaluate the significance of difference between the compared classifiers. Authors in the literature (Dietterich, 1998; Kuncheva, 2004) argue that McNemar is the most adequate statistical test in comparing classifiers' behavior.

In this thesis, we will analyze results using Micro/Macro averaging and compare different classifiers using both statistical tests: McNemar and paired samples Student's t-test in a coherent manner with state of the art works.

6 Testbed and Preliminary Experiments

This section presents our testbed details and first results obtained aiming to evaluate Rocchio, NB and SVM on three popular corpora. We chose to repeat these experiments on our testbed to have identical technical details for an unbiased comparison between the classifiers which is not possible using results from the literature. First and second subsections present some technical details on the classifiers and the corpora respectively. We identify also four measures for evaluating classification results. After a detailed discussion on results obtained from testing the classifiers on three corpora, we investigate the effect of training set labeling and organization on classification results.

6.1 Classifiers

In our experiments we use seven different classifiers: SVM, NB and five variants of Rocchio using five different similarity measures (see section 4). Here are some technical details for each of these classifiers:

- As for Rocchio: we implemented the classifier with the parameters as described in section 3.1. We use the Apache LuceneTM library for text indexing and we apply TF/IDF weighting scheme on resulting term frequency vectors. As for decision making, we implemented five different similarity measures (Cosine, Jaccard, Kullback Leibler, Levenshtein, Pearson) producing five different variations of the Rocchio classifier.
- As for NB classifier: we use its implementation in Weka (Hall *et al.*, 2009).
- As for SVM classifier: we use the package LIBSVM (Chang *et al.*, 2011) wrapped in WLSVM (EL-Manzalawy *et al.*, 2005) and integrated into the Weka environment (Hall *et al.*, 2009). We use the radial basis function as the kernel function.

6.2 Corpora

In these experimentations, we aim to evaluate the performance of Rocchio, SVM and NB on three different corpora: 20NewsGroups (Rennie), Reuters-21578 (Lewis *et al.*, 2004), Ohsumed (Hersh *et al.*, 1994).

6.2.1 20NewsGroups corpus

20NewsGroups corpus (Rennie) is a collection of 20,000 newsgroups documents almost evenly divided in twenty news classes according to their content topic assigned by authors. This collection is divided according to the percentages (60:40) into training corpus and test corpus respectively. Corpus organization in categories and the number of documents for each category in training and test sets are illustrated in Table 5.

Some classes cover similar topics for example (comp.sys.ibm.pc.hardware & comp.sys.mac.hardware), whereas others concern relatively different ones as (rec.autos & sci.crypt).

Category				Training	Test
Computer	comp	graphics		584	389
	comp	os	ms-windows	591	394
	comp	sys	ibm	590	392
	comp	sys	mac	578	385
	comp	windows	x	593	395
Sports	rec	autos		594	396
	rec	motorcycles		598	398
	rec	sport	baseball	597	397
	rec	sport	hockey	600	399
Forsale	mis	forsale		585	390
Science	sci	crypt		595	396
	sci	electronics		591	393
	sci	med		594	396
	sci	space		593	394
Politics	talk	politics	misc	465	310
	talk	politics	guns	546	364
	talk	politics	mideast	564	376
Religion	talk	religion	misc	377	251
	alt	atheism		480	319
	soc	religion	christian	599	398
Total				11314	7532

Table 5. Twenty actuality classes of 20NewsGroups corpus

6.2.2 Reuters

The corpus Reuters-21578 is a well-known dataset for text classification. The most used version as also confirmed in (Sebastiani, 2002) contains 12,902 documents for 90 classes, split up into test and training data (3,299 vs. 9,603) with the percentage 74,42% according to (Sebastiani, 2002). To obtain the Reuters 10 categories Apte' split (Sebastiani, 2002) we select the 10 top-sized categories listed in Table 6.

Category	Training	Test
acq	1650	719
corn	181	56
crude	389	189
earn	2877	1087
grain	433	149
interest	347	131
money-fx	538	179
ship	197	89
trade	369	117
wheat	212	71
Total	7193	2787

Table 6. Reuters-21578 corpus

6.2.3 Ohsumed

Ohsumed corpus (Hersh *et al.*, 1994) is composed of abstracts of medical articles of the year 1991 retrieved from the MEDLINE database indexed using MeSH (Medical Subject Headings). The first 20000 documents of this database were selected and categorized using 23 sub-concepts of the Mesh concept "Disease".

Category	Description	Training	Test
C04	Neoplasms	972	1251
C23	Pathological Conditions, Signs and Symptoms	976	1181
C06	Digestive System Diseases	588	632
C14	Cardiovascular Diseases	1192	1256
C20	Immune System Diseases	502	664
Total		4230	4984

Table 7. Ohsumed Corpus

The corpus is divided into Training and Test sets, so experimentations are realized in two phases: Training and Test. In this work, we restricted this corpus to the five most frequent classes (Yi *et al.*, 2009). For this dataset the split percentage is 42,30% according to (Joachims, 1998).

6.3 Testing SVM, NB, and Rocchio on classical text classification corpora

In these experimentations, three corpora are used: (i) 20NewsGroups (Rennie), (ii) Reuters (Sebastiani, 2002) (iii) OHSUMED (Hersh *et al.*, 1994). Each corpus is divided into Training and Test sets according to their corresponding references, so experiments are realized in two phases: Training and Test. Each of the seven classifiers is trained on the training set of each of these corpora in order to build the appropriate classification model. As for test, on each corpus, seven experiments are executed on the test sets (Holdout validation). For most classification

tasks, classifier's accuracy (Sokolova *et al.*, 2009) exceeded 90%. In order to evaluate system performance we use *F1-Measure*, *Precision* and *Recall* (Sokolova *et al.*, 2009) that give statistical information on the errors the classifiers make.

6.3.1 Experiments on the 20NewsGroups corpus

As illustrated in Figure 6, system's performance varies according to the classifier and the treated class. Results show SVM's superiority as compared with NB and Rocchio; SVM is more precise and makes fewer errors (Figure 6, Figure 7, Figure 8). Rocchio comes in the second place and NB in the last one.

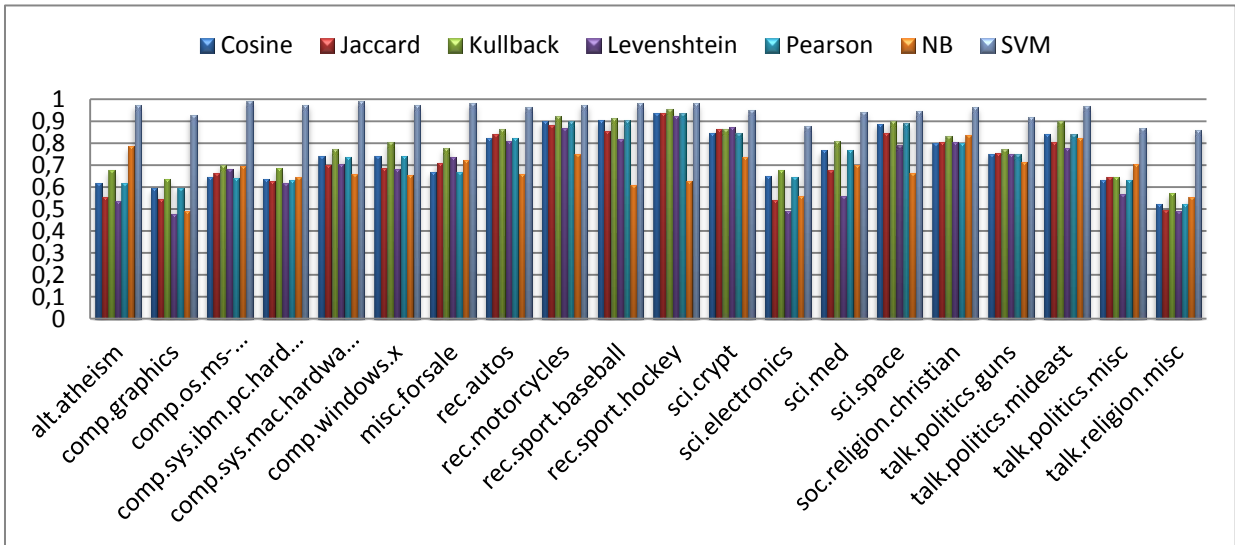


Figure 6. Evaluating Rocchio, NB and SVM on 20NewsGroups corpus using F1-measure

Although Rocchio comes in the second place after SVM, we can identify some critical issues that influenced its performance. For instance, the class "talk.religion.misc" is large compared to other religious classes. As observed in results, when a Rocchio classifier makes error classifying a document related to "talk.religion.misc", the resulting class is generally one of the religion related classes like "alt.atheism" (False negative). This explains the relatively low value of F1-Measure ranging between [0.5, 0.57] for "talk.religion.misc" that reflects a high precision and a low recall values (see Figure 6, Figure 7, Figure 8 respectively). We refer to such problem by problem of large classes.

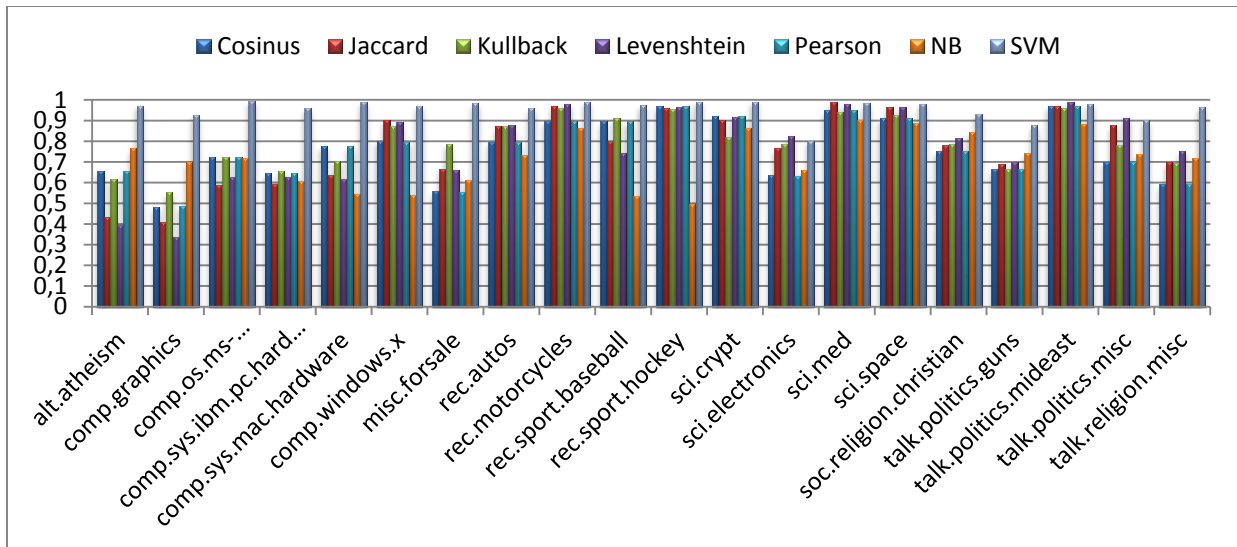


Figure 7. Evaluating Rocchio, NB and SVM on 20NewsGroups corpus using Precision

Another critical issue is related to similar classes. In this corpus, classes related to computers seem to use similar vocabulary which leads to similar centroids. By means of such centroids the classifier cannot distinguish classes properly (similar class issue) which results in F1-Measure values ranging from 0.5 to 0.8 in best cases. Nevertheless, all Rocchio-based classifiers perform well treating distinct classes like "rec.sport.hockey", "rec.sport.baseball" resulting in values that exceed (0.9).

After analyzing results in details, at least (50%) of incorrectly classified documents are classified in a similar class; this increases the False Negative for the right class and False Positive for the assigned class. Indeed, similar classes, using similar vocabularies, usually have their centroids close to each other in the feature space. This implies some classification difficulties in order to distinguish classes' boundaries affecting overall performance. In addition, document contents might be related to multiple classes making classifier's task tricky.

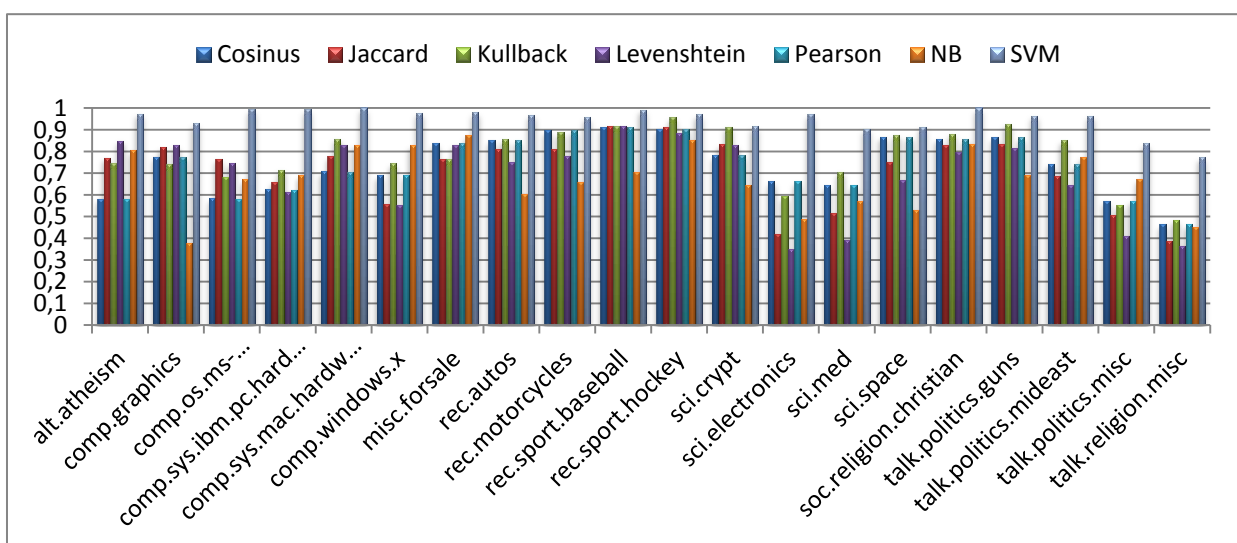


Figure 8. Evaluating Rocchio, NB and SVM on 20NewsGroups corpus using Recall

6.3.2 Experiments on the Reuters corpus

In these experiments, results show variations in performance which depends on the classification techniques and the treated class as well. As illustrated in Figure 9, NB seems to be the classifier with the worst results as it is the case in the previous test. The difference here is that SVM is not the best classifier since it shows some difficulties in classifying two classes (corn and wheat). Indeed, the general class “grain” covers both classes so SVM seems to recognize “grain” (high recall and low precision) and ignores “corn” and “wheat” which leads to zero values of F1-Measure, Precision and Recall for both classes (see Figure 9, Figure 10 and Figure 11 respectively). NB has the least classification effectiveness in this case.

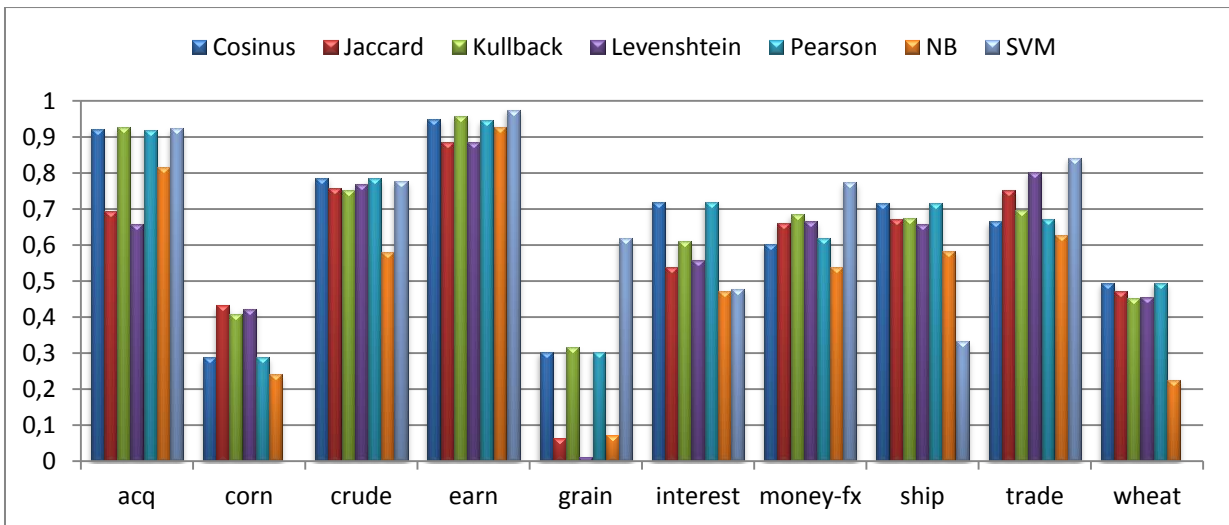


Figure 9. Evaluating Rocchio, NB and SVM on Reuters corpus using F1-measure

Rocchio shows some difficulties in classifying the general class "grain" as it contains information about both "corn" and "wheat" resulting in low F1-Measure (<0.5) as illustrated in Figure 9. Similarly to results on 20NewsGroups, this difficulty results in high precision and low recall for “grain” (Figure 10 and Figure 11 respectively). One can also observe similarities among classes like "trade" and "ship" that limit the F1-Measure to the maximum value of (0.8) whereas for more distinct classes the system attained (0.9) (example "earn" and "acq").

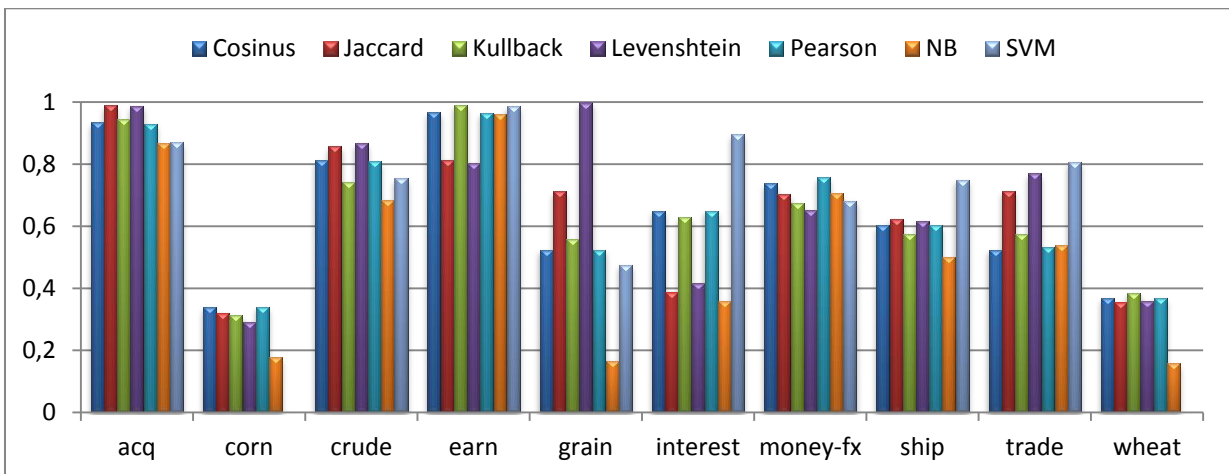


Figure 10. Evaluating Rocchio, NB and SVM on Reuters corpus using Precision

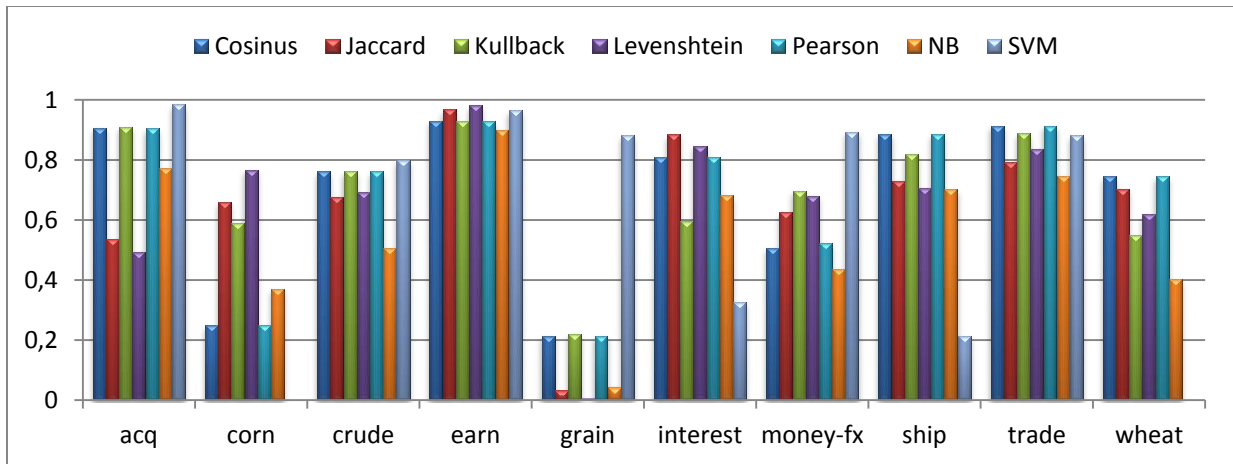


Figure 11. Evaluating Rocchio, NB and SVM on Reuters corpus using Recall

6.3.3 Experiments on the OHSUMED corpus

All classifiers demonstrate difficulties in classifying MEDLINE documents. Their results are very similar according to F1-Measure values in Figure 12. Observing Precision and Recall (see Figure 13 and Figure 14 respectively), we detect some variations in performance among these classifiers. For example, SVM seems to be more precise than other classifiers when tested on “C20” and “C06”. Nevertheless, SVM makes mistakes and attributes their documents to other classes which explain the relatively low recall values for both classes.

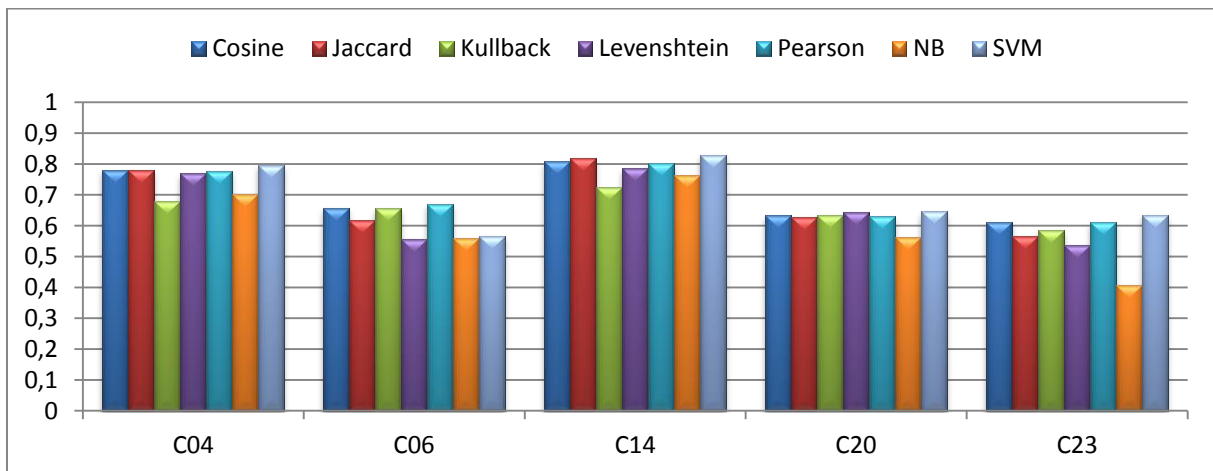


Figure 12. Evaluating Rocchio, NB and SVM on Ohsumed corpus using F1-measure

Similarly to previous experiments, NB classifier shows better results in few cases which has a slight or even no influence on its overall performance. According to Figure 14, NB has a slightly better recall value compared to other classifiers on “C20” but it also has the worst precision value on this class (see Figure 13) which results in a low F1-Measure value (<0.5) as illustrated in Figure 12.

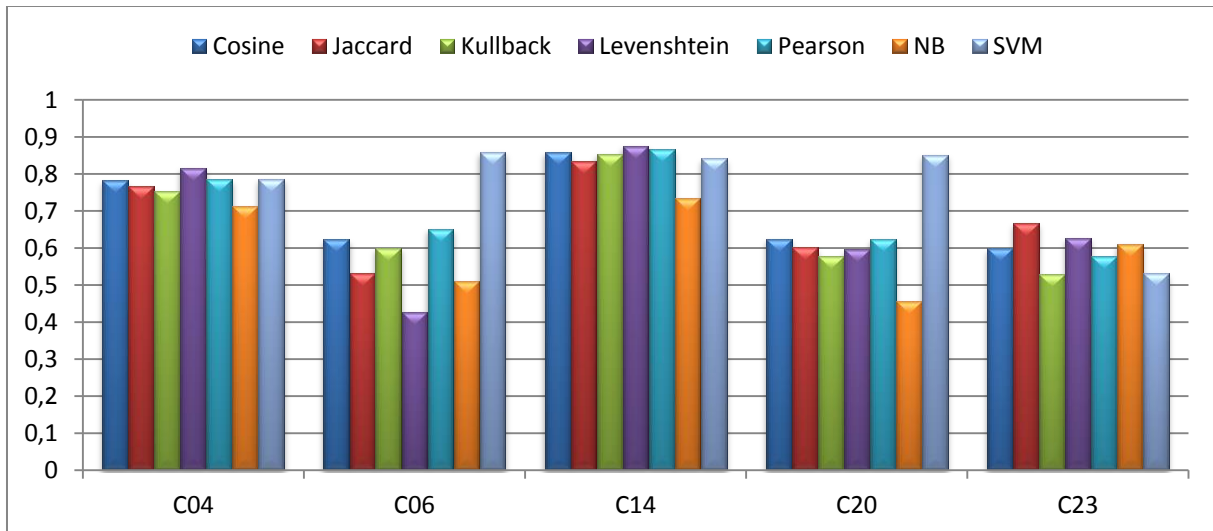


Figure 13. Evaluating Rocchio, NB and SVM on Ohsumed corpus using Precision

As for Rocchio classifiers, lowest results are for the class "C23" where pathology documents seem to be difficult to distinguish from other classes. In fact, this class is very large compared to others treated in the same case, and in other words, its documents can be related to other classes as pathologies can affect the digestive and the cardiovascular systems ("C06", "C14" respectively). As a result, low recall and F1-Measure values were observed for this class (≈ 0.5).

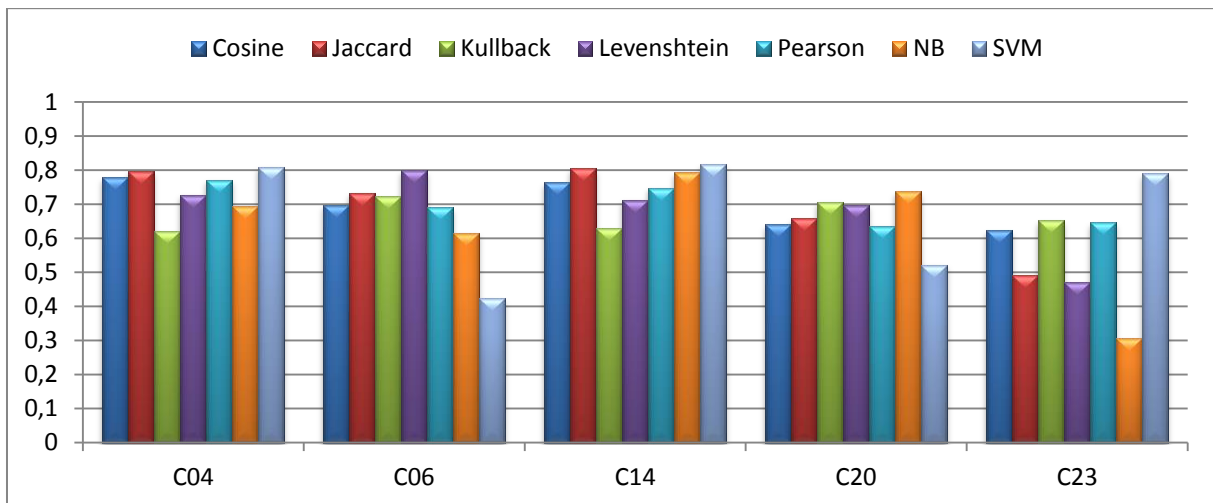


Figure 14. Evaluating Rocchio, NB and SVM on Ohsumed corpus using Recall

6.3.4 Conclusion

In this section, we tested seven classifiers: Rocchio with five different similarity measures, NB and SVM on three corpora: 20NewsGroups, Reuters and Ohsumed. Result analysis leads to these observations: First, classification results vary according to the classification technique in use and to the corpora contents and organization. Some classifiers like SVM demonstrate better results in some cases and critical limits in others. Second, General and large classes when mixed with other more specific classes are very difficult to recognize in most cases. Third, similar classes are very difficult to distinguish as they share a lot of characteristics among them.

Finally, domain specific documents like MEDLINE abstracts seem to be difficult to classify compared with general documents like actuality (20NewsGroups and Reuters). Thus we choose to investigate classification in the medical domain in the rest of this work.

Rocchio demonstrates stable performance in all tests (Albitar, Espinasse, *et al.*, 2012; Albitar, Fournier, *et al.*, 2012c) compared to SVM and NB, this makes it an adequate baseline for some approaches tested in this work and especially for advanced semantic integration. Next section investigates the effect of corpora organization on Rocchio's performance. We choose the 20NewsGroups for these tests since it is composed of twenty classes that cover both problems: general classes and similar classes.

6.4 The effect of training set labeling: case study on 20NewsGroups

In these experiments we try to investigate the effect of training set labeling and organization on Rocchio's classification results; to what extent large and similar classes can affect its performance. To answer these questions we try to highlight the difficulties identified earlier (large and similar classes) by modifying the 20NewsGroups corpus where Rocchio's performance was kind of poor compared with SVM (see Figure 6). We use two variations of the original corpus: (i) six chosen classes (distinct classes), and second (ii) the original corpus reorganized in six meta-classes according to Table 5; each group of similar classes are unified in one class. Rocchio then learns on each of these variations and calculates class centroid for each class of documents. As for test, Rocchio uses learned centroids with one of the five similarity measures on each variation of the corpus. We use F1-Measure (Sokolova *et al.*, 2009) for performance comparison.

6.4.1 Experiments on six chosen classes

In these experiments, we choose six classes, relatively distinct, of the twenty classes of the original corpus for both training and test. Classifier is first trained and then tested on the following classes: "comp.windows.x", "misc.forsale", "rec.auto", "sci.med", "soc.religion.christian", "talk.politics.mideast".

In general, Rocchio shows better performance on distinct classes; as their centroids are rather different and well dispersed in the feature space. Kullback-Leibler seems to outperform other similarity measures in these experiments as well. Results are illustrated in Figure 15 where columns follow the same order of legends from the left to the right.

Even though "sci.med" is treated with no other scientific classes for eliminating similar class problem, nevertheless Rocchio's performance is still relatively poor as compared with other classes. By observing results in Figure 15, Rocchio seems to do much better treating other classes like "comp.windows.x"; eliminating similar computer-related classes are more beneficial to classification than eliminating scientific ones. This is due to the large distribution of medical documents in the feature space so the learned centroid isn't an adequate prototype of the class.

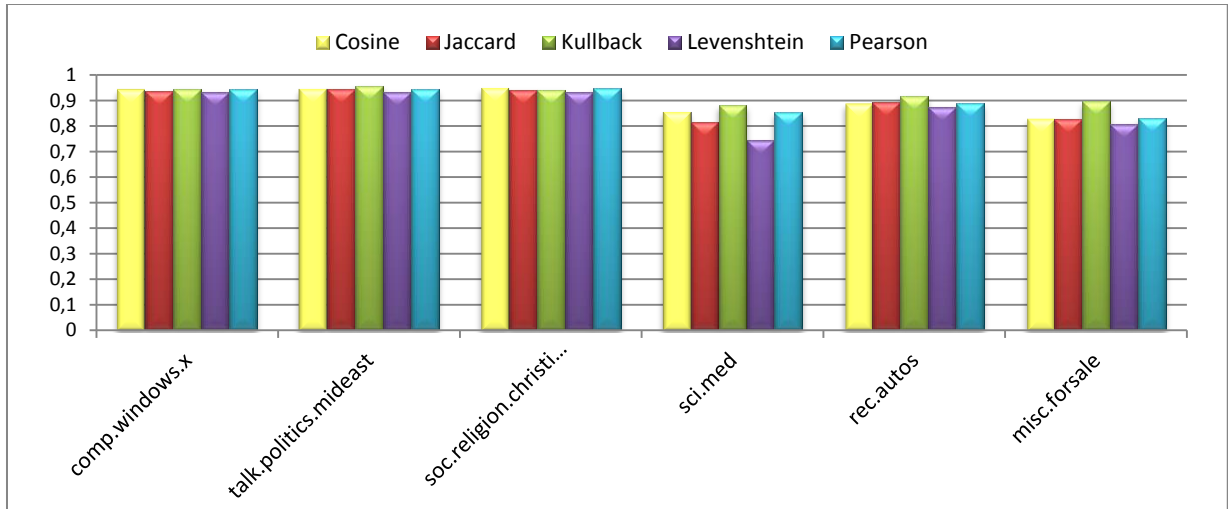


Figure 15. Evaluating five similarity measures on six classes of 20NewsGroups (F1-Measure)

6.4.2 Experiments on the corpus after reorganization

In these experiments, we reorganize the original 20NewsGroups corpus in six new classes in total: “comp”, “rec”, “science”, “forsale”, “politics”, and “religion”. As presented in Table 5 classes are reorganized depending on initial class similarities, so documents of similar classes are gathered in a more general class or a meta-class. Then, we train Rocchio on the training set to learn meta-class centroids that it needs along with different similarity measures for classifying the documents of test set according to these meta-classes.

According to results illustrated in Figure 16 (Columns follow the same order of legends from the left to the right), classifier's performance is relatively high for most classes, this applies at least to one of the similarity measures. In fact these classes assemble similar original classes like “religion” or well specified classes like “rec”. Classifier shows some difficulties classifying “science” as the classes it assembles contain diverse information (heterogeneous class issue). In fact, one centroid for such heterogeneous class is not very representative; this justifies the relatively poor value of F1-Measure for this class in Figure 16.

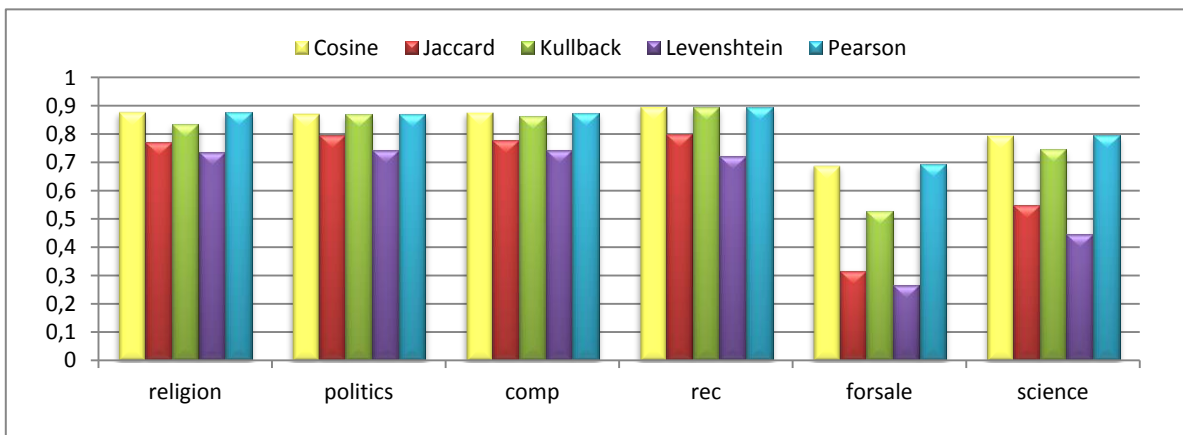


Figure 16. Evaluating five similarity measures on reorganized 20NewsGroups (F1-Measure)

6.4.3 Conclusion

In this section, we tried to assess the influence of training set labeling on different Rocchio-based classifiers in order to support our former observations and conclusions on large, similar and heterogeneous classes. We presented two supplementary tests, using in the first six distinct classes chosen from the original corpus, and in the second the original corpus reorganized in six meta-classes. We concluded that having similar classes, general classes or heterogeneous classes in the corpus can affect Rocchio's performance. Similarities among classes seem to have a relatively high influence on classification results.

In fact, Rocchio's limitations, as observed with similar classes, are mainly related to class representation and similarity calculations (Albitar, Espinasse, *et al.*, 2012). We propose to overcome them by means of semantic resources. We assume that by redefining centroids using concepts as terms we might limit intersections between spheres of similar classes in concept space. Consequently, ambiguities between classes using similar vocabulary can be resolved at representation level using semantic resources or ontologies.

Furthermore, documents related to specific domains like the medical domain need more attention since classical techniques seem to have difficulties in dealing with such documents, this is reason of our great interest in this domain.

7 Conclusion

This chapter is focused on text classification: origins, history and commonly used classical supervised techniques: Rocchio, SVM and NB. We tested and compared these techniques on three different corpora. SVM showed good results on 20Newsgroups compared to Rocchio and NB. However, it showed some difficulties on Reuters and much more on Ohsumed. Rocchio seems competitive with SVM especially when tested on Ohsumed. NB came always in the last place according to the results. We can conclude that the performance of the classifier depends on the context which makes it difficult to assign “*The Best Classification Technique*” to one of them.

Some limitations affected the performance of Rocchio in particular cases which led to investigate the effect of training set labeling on its performance. According to observations on results, some limitations seem to affect Rocchio's performance particularly when dealing with similarities among classes, general classes and heterogeneous classes. These limitations are mainly related to class representation and similarity assessment. We propose to overcome the limitations observed with similar classes by means of semantic resources; redefining centroids in the concept space might limit intersections between spheres of similar classes.

Despite its popularity, BOW seems to have some drawbacks like redundancy, ambiguities and orthogonality that we relate to the fact that BOW ignores semantics during text treatment. Therefore, vector-based representation (binary or TF/IDF) needs semantic enrichment using a certain background knowledge base (Hotho *et al.*, 2003) at the text representation level. We will investigate the influence of semantic text enrichment on classification using SVM, NB and Rocchio as well in chapter 5. Only Rocchio supports using knowledge bases in decision making through new semantic similarity functions (Guisse *et al.*, 2009). Its extendibility with semantic resources in the decision making process allowed us to apply advanced semantic integration through semantic similarity measures in chapter 5.

CHAPTER 3: SEMANTIC TEXT CLASSIFICATION

Table of contents

1	Introduction.....	53
2	Semantic resources.....	55
2.1	WordNet.....	55
2.2	Unified Medical Language System UMLS.....	56
2.3	Wikipedia.....	58
2.4	Open Directory Program ODP (DMOZ).....	59
2.5	Discussion.....	60
3	Semantics for text classification	62
3.1	Involving semantics in indexing	62
3.1.1	Latent topic modeling	63
3.1.2	Semantic kernels.....	64
3.1.3	Alternative features for the Vector Space Model (VSM).....	66
3.1.4	Discussion.....	70
3.2	Involving semantics in training	71
3.2.1	Semantic trees	72
3.2.2	Concept Forests	73
3.2.3	Discussion.....	73
3.3	Involving semantics in class prediction	75
3.4	Discussion.....	78
4	Semantic similarity measures	82
4.1	Ontology-based measures	82
4.1.1	Path-based similarity measures.....	82
4.1.2	Path and depth-based similarity measures	84
4.1.3	Discussion.....	86
4.2	Information theoretic measures	89
4.2.1	Computing IC-based semantic similarity measures using corpus statistics	89
4.2.2	Computing IC-based semantic similarity measures using the ontology	91
4.2.3	Discussion.....	92
4.3	Feature-based measures	95
4.3.1	The vision of Tversky.....	95
4.3.2	Feature-based semantic similarity measures	96
4.3.3	Discussion.....	99
4.4	Hybrid measures	101
4.4.1	Some hybrid measures	101
4.4.2	Discussion.....	103
4.5	Comparing families of semantic similarity measures.....	105
5	Conclusion	106

1 Introduction

In previous chapter we identified some challenging drawbacks in BOW used by traditional text classification techniques: dealing with redundancy related to synonymous features, resolving ambiguities by detecting the adequate meaning of polysemous words and finally considering semantic relations between words. So far, text classification is tackled from a statistical point of view which can be the origin of these limitations. We suggest that the intended meaning hidden in text must be involved in text classification towards a more effective semantic text classification.

According to the Cambridge Dictionary ("Cambridge Dictionaries Online, Cambridge University Press ", 2013), the notion of **Semantics** is "*the study of meaning in language*" so **Words** "*are semantic units that convey meaning*". A word with more than one meaning is **polysemous**; two words that have at least one meaning in common are said to be **synonymous** (Miller, 1995). A **Term** is "*a word or phrase used in relation to a particular subject*". Simple or complex terms denote a **Concept** in a particular context which is by definition "*a principle or an idea*". Many research works focus on how to structure, classify, model, and represent the concepts and relationships concerning a particular domain of interest (Astrakhantsev *et al.*, 2013). Having an agreement on a semantic resource enables researchers to share and use this resource in a way that is consistent with its specification (Gruber, 1995). For example, synonymous terms are used in the same way according to the provided definition avoiding all ambiguities. Furthermore, controlled vocabularies can be reusable and cross lingual as well.

Controlled vocabularies are the broadest category of semantic resources which includes: taxonomies, thesauri, ontologies, etc. The main differences between these kinds are:

- How much meaning is attributed to concepts.
- How this meaning is noted in the concepts and the relations between them.
- How the vocabulary is used.

A controlled vocabulary may have no meaning or a specific meaning for each term. **Taxonomies** put the vocabulary in a hierarchical structure with generalization/specialization relations that are usually referred to by "*is a kind of*". This makes taxonomies an adequate "*system for naming and organizing things, especially plants and animals, into groups that share similar qualities*" ("Cambridge Dictionaries Online, Cambridge University Press ", 2013). A **Thesaurus** is "*a type of dictionary in which words with similar meanings are arranged in groups*". Thesauri use another type of relation in addition to broader/narrower which is similar to the one used in Taxonomies. This additional relation is referred to using different names like synonym of, similar to, related to etc.

By definition, **Ontology** is "*the part of philosophy that studies what it means to exist*" ("Cambridge Dictionaries Online, Cambridge University Press ", 2013). This notion is adopted and refined for information science in 1990s by **Gruber** as the following: "*Ontology is an explicit specification of a shared conceptualization which is in turn the objects, concepts, and other entities that are presumed to exist in some area of interest and the relationships that hold among them*" (Gruber, 1995). It is notably remarked that ontology is used to refer to previous

kinds of controlled vocabularies as well despite the differences among them. In fact, ontology is a model of the knowledge related to a particular domain that supports reasoning about its concepts. Ontologies are mainly used in artificial intelligence (Dobrev *et al.*, 2008), the Semantic Web (Trillo *et al.*, 2011), software engineering (Wongthongtham *et al.*, 2009), medical informatics (Meystre *et al.*, 2010), etc.

Next section presents some semantic resources already used in semantic text classification in some details. Section 3 presents different state of art approaches involving semantic knowledge in text classification and similar tasks related to IR. These approaches deploy different semantic resources at different steps in the process of text classification: text representation, training and in classification as well. Section 4 presents a state of art semantic similarity measures that assess the semantic similarity between pairs of concepts in the semantic resource. This semantic similarity is deployed in many state of the art approaches presented in section 3 in order to involve semantics in text classification.

2 Semantic resources

The major interest of research on semantics is to provide semantic resources or controlled vocabularies that cover different domains of interest. These resources provide a concept consensus through term normalization and disambiguation in a particular context which facilitates intra-lingual and cross-lingual knowledge sharing.

Research on semantics gave birth to general semantic resources like WordNet® (Miller, 1995), Yago (YAGO, 2013), SUMO (SUMO, 2013), etc. Some researchers were interested in developing domain specific semantic resources like UMLS® (2013) for the medical domain and AGROVOC (AGROVOC, 2013) that covers all areas of interest to FAO, including food, nutrition, agriculture, fisheries, forestry, environment etc. In addition to research, collaborative work on the Web introduced other useful general resources like Wikipedia (2013), *Open Directory Program* (ODP) (2013), etc. In fact, such collaborative projects implicate internet users in archiving and organizing information on the Web.

Semantic text classification is one of the application fields where semantic resources are deployed intensively. Next subsections present in details the most commonly used resources in this field: WordNet, UMLS, Wikipedia and ODP.

2.1 WordNet

WordNet® (Miller, 1995) is a lexical database for English developed to be deployed under program control. In other words it adapts traditional lexicographic information for modern computing. George A. Miller directed the development of WordNet starting from 1985 at Princeton University heading a team of psycholinguists and linguists to test psycholinguistic theories on how humans use and understand words. WordNet has become a large computer-readable electronic lexicon deployed in applications such as IR (Boubekeur, 2008), text classification (Séaghdha, 2009), sense disambiguation (Navigli, 2009) and so on.

WordNet covers the majority of nouns, verbs, adjectives and adverbs of English structured in a network of nodes and links. Each node, called synset (SYNonym SET), consists of a set of synonyms. Synonyms that have the same meaning are grouped together at a node to form a synset that conveys a particular sense of a distinct concept. Each synonym is a simple or a complex term: one word or a group of words respectively.

WordNet synsets are connected by links or semantic relations that go beyond those of a classical thesaurus. The basic relationship between the terms of the same synset is synonymy. The different synsets are otherwise bound by various semantic relations such as the following (Miller, 1995):

- Synonymy (symmetric) is WordNet's major relation since WordNet organizes terms sharing the same meanings (or synonyms) in synsets.
- Antonymy or opposing-name (symmetric) is essential to organize the meanings of adjectives and adverbs.
- Hyponymy (sub-name) and its inverse, hypernymy (super-name), are transitive relations between synsets that organizes the meanings of nouns into a hierarchical structure so

general concepts are hypernyms of more specific concepts. For example (Figure 17), “canine” is a hypernym of “dog”, “wolf” and “fox”.

- Meronymy (part-name) and its inverse, holonymy (whole-name), are complex semantic relations that hold between a whole (holonym) and its parts (meronyms). “car” has meronyms “engine”, “wheel”, etc.
- Troponymy (manner-name) is to organize verbs in a hierarchy like “walk” and “step”.
- Entailment relations are especially between verbs like causality between “show” and “see”.

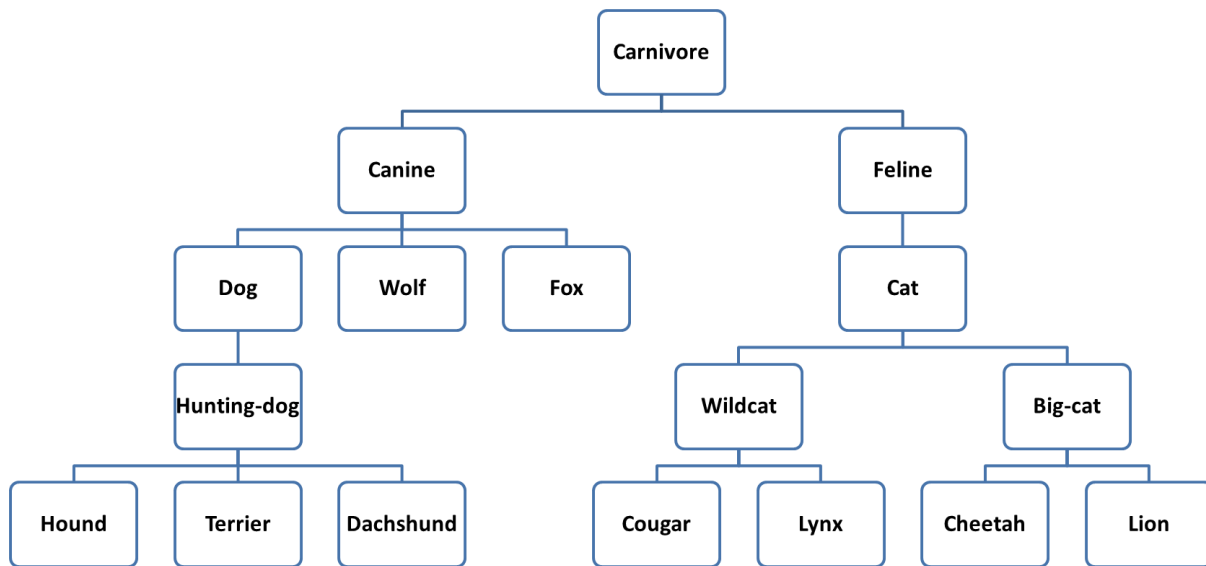


Figure 17. Part of WordNet with hypernymy and hyponymy relations.

WordNet contains a wide range of common English words (WordNet 3.0 counts 147278 terms) which is its major advantage. Nevertheless, it does not cover special domain vocabulary like the medical domain. Thus, it is proved to be useful in treating information related to general domains like actualities which is not true for application in uncovered domains where it is necessary to use domain ontology.

2.2 Unified Medical Language System UMLS

The *Unified Medical Language System* (UMLS®) was developed at the *National Library of Medicine* (NLM) in the intent to model the language of biomedicine and health and to help computers understand the language of medicine. In fact, UMLS knowledge sources enhance the development of information systems in the medical domain. The UMLS knowledge base consists of three main resources: the Metathesaurus, the Semantic Network and the SPECIALIST Lexicon (Figure 18).

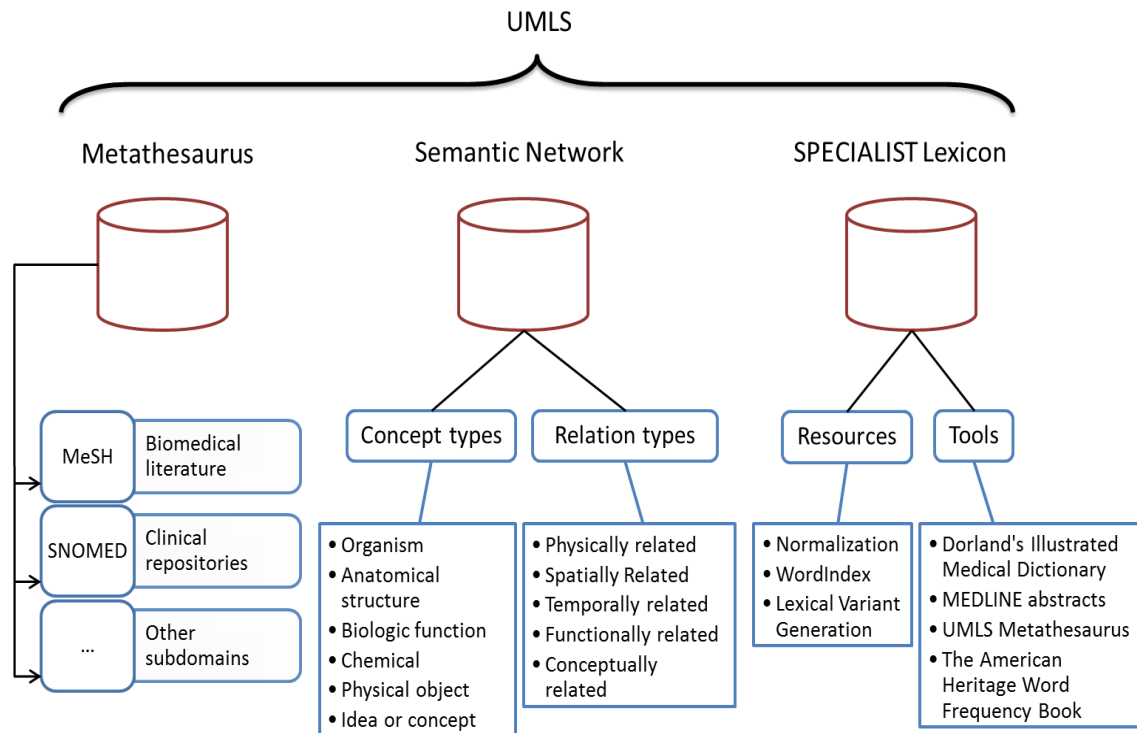


Figure 18. The various resources and subdomains unified in UMLS

The *Metathesaurus* is a multilingual vocabulary database of medical concepts, their names, their attributes and the relations among them. This database gathers concepts of the various source vocabularies according to their senses, grouping synonym terms together under a unique concept. In the *Metathesaurus*, each *concept* has a unique identifier, a name, at least one semantic type from the semantic network and at least one definition. The *relations* among concepts are either *structural* (hierarchical) or *associative*. From the semantic resources unified in the *Metathesaurus* (see Figure 18), we mention particularly the MeSH thesaurus (*Medical Subject Headings*) and SNOMED-CT terminology (*Systematized Nomenclature Of Medicine Clinical Terms*).

Concepts and relations among them are assigned at least one type from the *Semantic Network*. Indeed, the *Semantic Network* provides a higher level of abstraction through concept and relation categorization in inter-related types constituting a network of 133 semantic types and 54 relationships. The detailed specific information of concepts are located in the *Metathesaurus* while the *Semantic Network* provides semantic and relation types that can be affected to concepts like (Organism, Anatomical structure, Biologic function, etc.) and to the relations among them like (Physically related, Spatially Related, Temporally related, etc.). Figure 18 illustrates some of these types.

The *SPECIALIST* lexicon contains a large variety of general words that are retrieved from different resources like *The American Heritage Word Frequency Book*. In addition, it contains words related to the medical domain that are retrieved from a variety of resources like *Dorland's Illustrated Medical Dictionary*, MEDLINE abstracts, and UMLS *Metathesaurus*. The *SPECIALIST* assembles syntactic, morphological (inflection, derivation, and composition) and

orthographic (spelling) information for each word which is used by lexical tools for Natural Language Processing (NLP) like Normalization, WordIndex and Lexical Variant Generation.

2.3 Wikipedia

Since concepts are the elementary units of knowledge, encyclopedias like Wikipedia® can be used as eligible sources of concept knowledge. In fact, they give detailed description of each concept in addition to a relatively rich linking with related concepts.

Since its creation in 2001, Wikipedia counts more than 22,000,000 articles in 285 languages (4,295,594 articles in English) thanks to more than 77,000 active contributors (2013). These articles cover extensive concepts in all branches of knowledge, and provide factual description of the concepts (Hartmann *et al.*, 1998). Each article contains hyperlinks to related articles that provide a sort of semantic relations among the concepts they describe. Figure 19 illustrates the articles in English Wikipedia for the concept “Classification”.

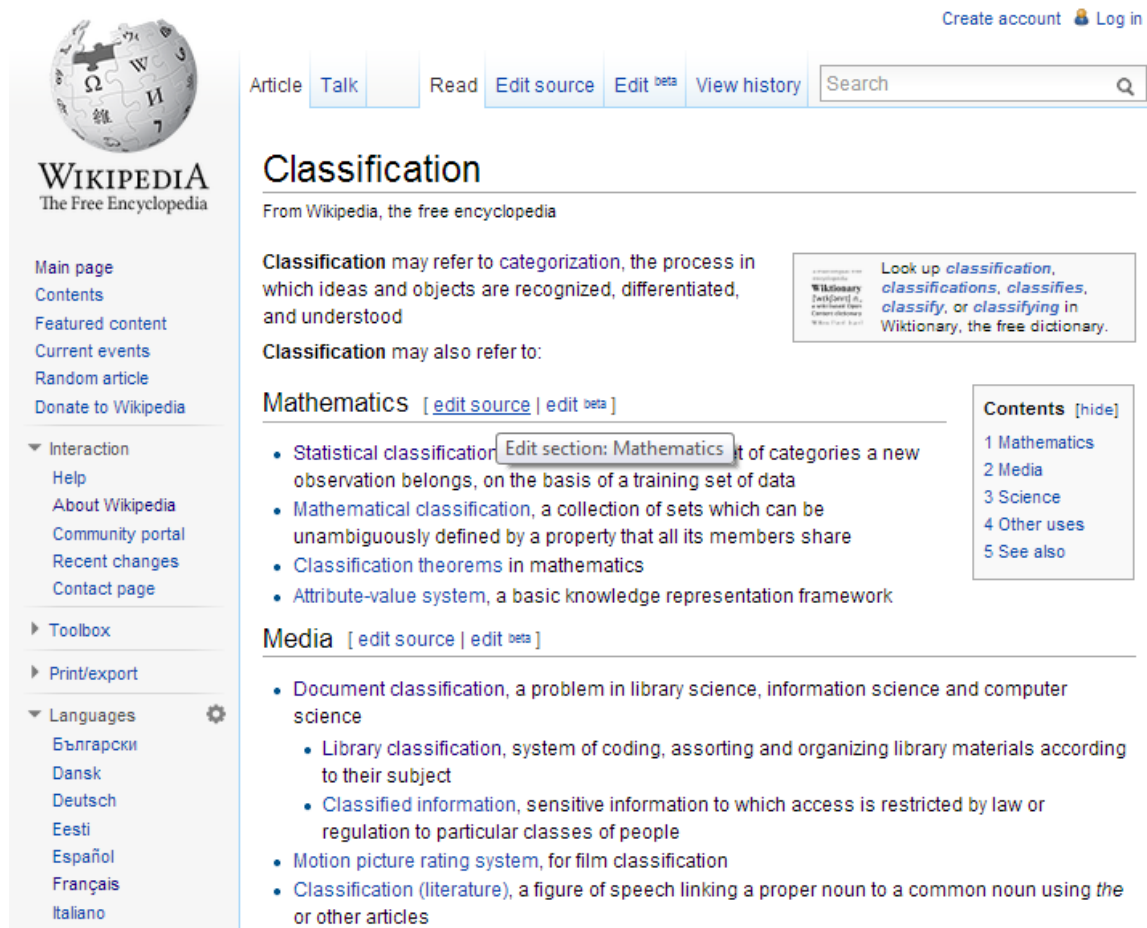


Figure 19. Wikipedia: Page for “Classification” with links to different articles related to different languages, domains and contexts of usage.

Wikipedia’s open accessibility and comprehensive world knowledge encouraged researches to use it as an effective semantic resource in many challenging tasks related to text processing like information retrieval (D. N. Milne *et al.*, 2007), text categorization (Gabrilovich *et al.*, 2007; Wang *et al.*, 2008), and in text clustering (L. Huang *et al.*, 2012).

All studies using Wikipedia as a semantic resource consider articles as concepts and link words and phrases in text to these articles according to their intended meaning. Polysemous words can be mapped to multiple articles according to their different meanings in different contexts such as those shown in Figure 19 for the concept “Classification”. In such cases the contextual information of articles in Wikipedia is compared with the context of the treated word to find the best match and to resolve the ambiguity. The unique identifiers of the mapped articles in Wikipedia are used as features in text representation. Some researchers derived a vocabulary from Wikipedia articles to provide a well-structured semantic resource to their semantic applications (Mihalcea *et al.*, 2006; D. Milne *et al.*, 2008).

2.4 Open Directory Program ODP (DMOZ)

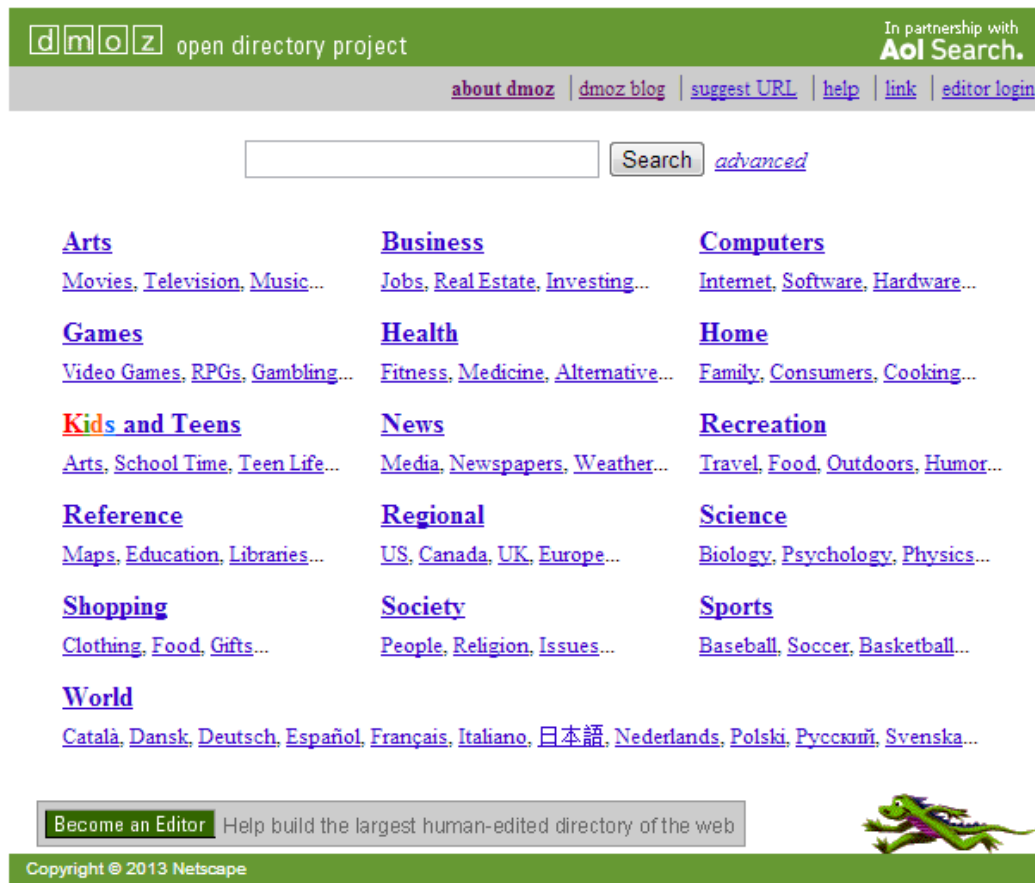
The Open Directory Program (ODP©) (2013), more known as Dmoz, is a websites directory founded in 1998 by Rich Skrenta and Bob Truel in California, U.S.A as an open-content directory edited by volunteers. The ODP is considered the largest and the most comprehensive directory on the Web edited by a large community of volunteers from all around the world. It lists now nearly five million sites thanks to more than 90000 volunteers. The ODP uses a hierarchical structure to organize lists of Web sites. Semantic concepts or similar topics are grouped into categories which may have subcategories as well.

The ODP is constructed manually by users of the Web by associating web pages with the most similar category of concepts or topic in the ODP. Each concept in ODP represents a topic of interest to Web users that is defined by a title and a description that summarizes contents of the associated Web pages.

The concepts of ODP are interconnected by semantic relations such as "is-a", "symbolic" and "related to":

- The relation (is-a) organizes the concepts in a hierarchy from the more general to the more specific concepts.
- The relation (symbolic link) is a hyperlink that connects a Web page to another one in the same directory. Symbolic links enables the editors to establish shortcuts between web pages in a directory, and also to attribute the web page multiple categories.
- The relation (related-to): to point to other semantically related concepts. For example (see Figure 20), “operating system” is a “software” which is related to “computers”.

ODP is mainly used in applications related to user profiles and personalization of IR. User profiles can be constituted of concepts of the ODP related to the web pages visited by the user (Chirita *et al.*, 2005). Then, the constituted profile is used to re-rank web pages that are retrieved by a classical IR system to personalize its results according to the topics of interest of the user (Daoud, 2009). Despite the size of information coded in ODP, ODP is based on what people are looking for on the Web, and how they search the Web for information which makes it different from other semantic resources.



5,223,457 sites - 98,565 editors - over 1,019,020 categories

Figure 20. ODP home page. General concepts are in bold (2013).

2.5 Discussion

This section presented four state of the art semantic resources that have been used in semantic text processing for different applications: WordNet, UMLS, Wikipedia and ODP. These resources are compared in Table 8.

Resource	Origin	Domain	Principle components	Limitation
WordNet	Research	General	Synsets and relations	Specific domains are uncovered
UMLS	Research	Biomedical	Concepts and relations	Domain specific
Wikipedia	Collaborative	General	Interlinked articles	Specific domains are uncovered
ODP	Collaborative	General	Web pages associated with interlinked concepts	Lack of semantics

Table 8. Comparing four semantic resources: WordNet, UMLS, Wikipedia and ODP.

Both WordNet and UMLS are ongoing research projects aiming at complete and large electronic knowledge bases that can be deployed by computers for a better text understanding. On the contrary, Wikipedia and ODP are the result of a collaborative work of internet users. Wikipedia provides millions of articles in different languages on all branches of knowledge and ODP intends to organize the information of the Web under categories and concepts. In fact, ODP can

be used as a source of concept knowledge as it provides a larger coverage nevertheless it is less effective than other well-structured rich semantic resources (L. Huang, 2011). Wikipedia encodes rich semantic relations among concepts as compared to ODP which is particularly useful for sense disambiguation (Mihalcea, 2007).

Rich semantic resources like WordNet are very effective and useful in text classification. Since concepts in WordNet are generic some specific domains aren't well covered which implies the use of domain-specific resources in text processing in such domains (Hotho *et al.*, 2003; Zhu *et al.*, 2009).

3 Semantics for text classification

Typically, most of supervised text classification techniques are based on statistical and probabilistic hypothesis in both training and classification procedures. As for text representation or indexing, the importance of a term to a document is assessed using the frequency of its occurrences in the document. So far, the intended meaning of terms and the relations among them are not treated or used in text classification. In other words, semantics and relatedness behind literally occurring words are missing in classical text classification techniques as presented in previous chapter. The question being raised is: *Does semantic information help in the text categorization task?* (Ferretti *et al.*, 2008).

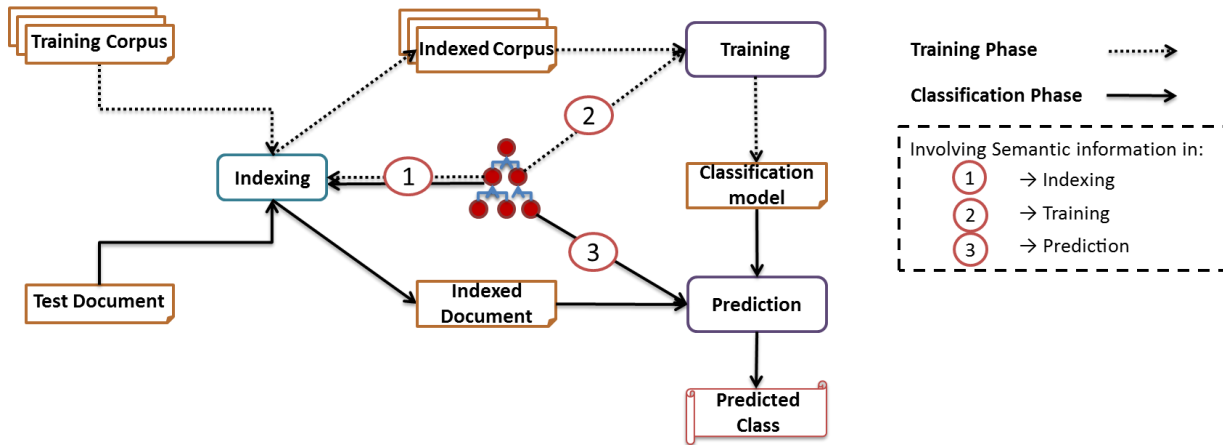


Figure 21. Involving semantic resources in supervised text classification system: a general architecture

In this thesis, we aim to answer this question or at least to determine where and how semantics are useful to text classification and to what extent it can help in better classification. The three possibilities (see Figure 21) were investigated by multiple works in the literature. We will survey these works and discuss their limitations in next sections. First, semantic resources may be useful at text indexing step so index would contain words, phrases, concepts or a combination of these forms. Moreover, implicit semantics that can be discovered through latent topic modeling approaches can also be used in text representation. Second, they can help in learning the classification model, or the model might be based on concepts and their relatedness so semantics are involved during the training step as well. Third, semantics can also be useful in class prediction.

3.1 Involving semantics in indexing

To involve semantic features in indexing (Figure 21, arrow 1), state of the art approaches used either implicit semantics through topic modeling or explicit semantics derived from structured resources or controlled vocabularies and used as new features for text representation. Other approaches use either types in semantic kernels to support some supervised classification techniques. Next subsections detail some popular approaches for: Latent topic modeling, Semantic Kernels and alternative features for the VSM.

3.1.1 Latent topic modeling

Latent topic modeling is a family of statistical techniques that extract implicit topics or concepts from texts by deriving lists of co-occurring words using text statistics. The basic hypothesis is that the words constituting topics co-occur in meaningful ways so by identifying these topics semantics are injected into the BOW's vocabulary. These techniques have many similarities. They use the BOW model as a starting point and then they reduce its dimensionality to concepts or topics with a weighted list of terms for each concept and weighted list of concepts for each document (Crossno *et al.*, 2011).

Three well-known mathematical techniques for text documents modeling appealed the interest of many researchers in the information retrieval domain (Liu *et al.*, 2004; Mitra *et al.*, 2007; Somasundaram *et al.*, 2012; Deveaud *et al.*, 2013). We present three major approaches: *Latent Semantic Analysis* or *Indexing* (LSA or LSI respectively) (Deerwester *et al.*, 1990), *Probabilistic Latent Semantic Analysis* (pLSA) (Hofmann, 1999) and *Latent Dirichlet Allocation* (LDA) (Blei *et al.*, 2003).

LSA (Deerwester *et al.*, 1990) uses *Singular Value Decomposition* (SVD) to discover implicit higher-order structure in the co-occurrences of terms within documents. In fact, this technique projects the large sparse matrices representing documents in the VSM into a subspace limited to the largest singular vectors of these matrices. This subspace is known by *Latent Semantic Space*. LSA aimed to overcome the limits of lexical matching in classical VSM-based techniques and especially the synonymy and polysemy problems. Implicit concepts that are statistically discovered by LSA prove to be more relevant to indexing than literally occurring words in many applications (Berry *et al.*, 1995).

pLSA (Hofmann, 1999) evolved from LSA and is considered as a probabilistic variant since it uses the likelihood principle instead of SVD for dimensionality reduction. Each document representation is reduced to a probability distribution on a fixed set of implicit topics or concepts (Blei *et al.*, 2003) resulting in a list of numbers of the different proportions for topics.

LDA (Blei *et al.*, 2003), is also based on a probabilistic model. It uses a generative approach on three hierarchical levels: documents, topics and words of the collection vocabulary. Documents are represented as random mixtures of topics. A topic has probabilities of generating various words which are learned on the collection.

The three previous techniques are considered as feature transformation methods; they generate a new smaller set of features as a function of the original set in the feature space (Liu *et al.*, 2004; Mitra *et al.*, 2007; Somasundaram *et al.*, 2012). Nevertheless, since these techniques are unsupervised, they are not adapted to supervised text classification. In fact, they ignore the underlying class-distribution of the training corpus and try to suggest the best class distribution of the documents according to the generated features. As the features found by these techniques are not necessarily compatible with the class distribution of the corpus, the quality of results is not guaranteed (Liu *et al.*, 2004; Aggarwal *et al.*, 2012).

A number of extended versions of these techniques have also been proposed to overcome the previous limitations by using the class labels for effective supervision. Authors in (Liu *et al.*, 2004) proposed *Supervised Latent Semantic Indexing* (SLSI). In fact, they apply LSI

iteratively to subsets of the corpus, each corresponding to a particular class, in order to identify the discriminative features of each of the classes. After creating class-specific LSI feature sets or local sets, test documents are compared against each set in order to create the most discriminative reduced set for representing these documents. The major drawback of this adaptation of LSI to supervised text classification is that different feature sets are in different subspaces, and therefore it is difficult to compare documents across these subspaces. Furthermore, this approach tends to be computationally expensive as compared to the relatively low gain in classification quality on Reuters-215783 and Industry Sector4 (Liu *et al.*, 2004; Aggarwal *et al.*, 2012).

In fact, Latent topic modeling approaches are effective methods for text representation using extracted implicit semantics. Nevertheless, these approaches are inherently unsupervised which makes their adaptation to supervised text classification delicate and harmful to their efficiency and effectiveness.

3.1.2 Semantic kernels

Considering Kernel based classifiers like SVM, many works propose using semantic-based kernel functions which are also known by *Semantic Kernels*. The source of the semantics can be based on term co-occurrences in the collection. In this case, the classifier uses the former family of techniques resulting in distributional kernels. Another source is semantic similarities between terms that can be derived from a particular knowledge base like encyclopedia, taxonomies and ontologies to generate semantic similarity kernels. In semantic kernels, the feature space is a concept space constituted of concepts from the used semantic resources. In other words, original document vectors are projected into the concept space through word to concept mapping which will be discussed in next section.

Authors in (Séaghdha *et al.*, 2008) used observed co-occurrences in the collection of documents to construct distributional kernels for SVM-based classifiers. These classifiers were applied to three different tasks: compound noun interpretation, identification of semantic relations among nominals in text and verb classification. Authors (Séaghdha *et al.*, 2008) also reported that distributional kernels with co-occurrence probability distributions are suitable for different semantic classification problems and can improve the performance of SVM more than other classical kernels. This approach was tested on the identification of semantic relations among nominals in text which is task 4 of SemEval competition (Girju *et al.*, 2007).

Authors in (Séaghdha, 2009) used WordNet to construct semantic kernels of similarities that use WordNet's noun hierarchy (Miller, 1995) as a graph with hyponymy relations and a similarity measure (Séaghdha, 2009). Authors reported that SVM works better with semantic kernels when applied to the identification of semantic relations among nominals in text (Girju *et al.*, 2007). Authors used WordNet as an explicit semantic resource with semantic similarities for the semantic kernel instead of the probabilities of co-occurrences of the distributional kernel used in the previous one.

Authors in (Bloehdorn *et al.*, 2007) take advantage of the linguistic structures like syntactic dependencies of text and combines it with a WordNet-based semantic similarity between terms to constitute a semantic kernel using different semantic similarity measures.

Authors reported improvement in SVM’s performance in *Question Classification* domain (QC) on TREC datasets. In fact, accurate question classification according to different types is essential to locate and extract the correct answer.

Other works enriched text representation by deploying the embedded knowledge in encyclopedia like Wikipedia (2013) which may be very effective resource for concept knowledge. Compared with WordNet, Wikipedia resolves ambiguities and provides associative relations between concepts as well (P. Wang *et al.*, 2007).

Authors in (P. Wang *et al.*, 2007; Wang *et al.*, 2008) used derived semantics from Wikipedia to construct semantic kernels. These kernels are used to enrich text representation with conceptual information and can enhance the prediction capabilities of classification techniques. Effectively, authors reported that Wikipedia-based semantic kernels helped SVM in classification (Wang *et al.*, 2008) when tested on Reuters-21578 (Lewis *et al.*, 2004), Ohsumed (Hersh *et al.*, 1994), 20NewsGroups (Rennie, 2013) and Movies . The semantic kernel in this case is a semantic similarity matrix that compares features or terms from the feature space pair-to-pair using a particular semantic similarity measure. In fact, applying semantic kernel to document vector representation seems to make resulting representation less sparse.

For example (Wang *et al.*, 2008), given two document term vectors in Table 9 and a semantic similarity matrix in Table 10. Using a simple inner product as a kernel function the enriched term vectors are given in Table 11. For example in original vectors (see Table 9), the term “Puma” occurred two times in d_1 whereas neither “Cougar” nor “Feline” occurs in d_1 . On the other hand, only the term “Cougar” appears in the document d_2 . As the documents share no term in common, direct lexical matching will result in zero similarity between these documents. On the contrary, after applying the semantic kernel to these vectors they become less sparse as the frequency of each term can be propagated to similar terms according to its similarity with them in the semantic matrix.

	Puma	Cougar	Feline	...
d_1	2	0	0	...
d_2	0	1	0	...

Table 9. Two documents (d_1, d_2) term vectors. Numbers are term frequencies in document

	Puma	Cougar	Feline	...
Puma	1	1	0.4	...
Cougar	1	1	0.4	...
Felin	0.4	0.4	1	...
...	...	...	...	...

Table 10. Semantic similarity matrix for three terms: Puma, Cougar, Feline.

Obviously, resulting vectors are less sparse from the original ones as similar terms are taken into consideration in addition to literally occurring ones. This helps classification techniques enhance their performance as they involve semantics in text representation. The main drawback of this approach is that adding concepts to text representation might affect the effectiveness and

the efficiency of classification. Authors used heuristics to limit the added concepts to the N most similar concepts.

	Puma	Cougar	Feline	...
d_1	2	2	0.8	...
d_2	1	1	0.4	...

Table 11. Two documents (d_1, d_2) term vectors. Numbers represent weights after inner product between a line from Table 9 and a column from Table 10.

Domain ontologies were also used in constructing semantic kernels. Authors in (Aseervatham *et al.*, 2009) used UMLS, a well-known ontology in the medical domain, to construct their semantic kernel. In fact, authors reported improvements in SVM-based classification using this approach as compared to other kernel functions when applied to medical documents. The ambiguities are not treated in this work which is considered its main limitation.

Originally, kernel functions projects classification into a new feature space where training examples can be linearly separable. This helps SVM learn classification models effectively. Many state of the art works investigated the role of semantics in building semantic kernels and their effects on SVM classifiers. Yet, to the best of our knowledge none of these works applied semantic kernels to other classification techniques.

3.1.3 Alternative features for the Vector Space Model (VSM)

Many works proposed new extensions to the classical BOW in order to overcome its limitations that we investigated in (see chapter2 section 2.6). Numerous weighting schemes for the classical BOW are proposed in (Lan *et al.*, 2009), all aiming to optimize feature weights in the original BOW, which might improve text classification. Moreover, other works demonstrated some improvements by introducing new features to the original BOW. In this section, we present a survey on these works to identify different features by which they extended the classical Bag of Words model (BOW). Next subsections present phrases and concepts as alternative features for text representation.

3.1.3.1 Phrases

Terms used in the classical BOW model may co-occur in text in particular contexts. This implies that co-occurring terms might convey meaning as well as single terms and they may be useful in text representation. Authors in (Caropreso *et al.*, 2001; Z. Li *et al.*, 2009) propose a **Bag of Phrases** (BOP) model instead of the classical *Bag of Words* (BOW) taking frequently occurring N-gram phrases into account during indexing. Since early 90s, many works proposed using Bag of Bigrams for text representation. Authors in (Caropreso *et al.*, 2001) compared the use of unigram (single terms) with the use of bi-grams (two-words terms) in text representation and evaluated its effect on classification using SVM and Rocchio. Using different feature selection methods, authors studied how including bigrams in text representation can affect text classification on the corpus Reuters-21578. Nevertheless, authors reported deterioration in classification effectiveness when bigrams are excessively used at the expense of unigrams.

Authors in (Z. Li *et al.*, 2009) argue that using phrases in the text representation model is beneficial to text classification especially for similar texts. Usually, similar texts nearly use the same word set thus it is difficult to distinguish them using the classical BOW. Nevertheless, each of the similar topics has its respective set of phrases that helps the classifier to enhance its capabilities. According to tests on KNN, Decision Trees, SVM and NB using a collection of database related papers from ACM digital library, the proposed BOP outperforms the original BOW (Z. Li *et al.*, 2009).

For example, “Text Mining” and “Data Mining” are similar topics that share the word set {text, data, mining}. However, “text mining” and “data mining” are specific to the topics respectively; they aren’t commonly used in both topics. That means that adding “text mining” and “data mining” to the bag for text representation might help in distinguishing the similar topics. Other studies demonstrated marginal improvement or a certain decrease when representing texts from different fields using bags of N-grams.

Despite the improvements demonstrated in some works using BOP (Caropreso *et al.*, 2001; Z. Li *et al.*, 2009), few of BOW's limitations are treated. Furthermore, BOP is sparser than BOW and also suffers from ambiguities although phrases in general are more specific than single terms (Stavrianou *et al.*, 2007; L. Huang *et al.*, 2012).

3.1.3.2 Concepts

Concepts are considered the best alternative features as they encompass the three drawbacks of BOW related to synonymous and polysemous words and the use of relations and similarities among words in the model. Thus the original BOW can be transformed into **Bag of Concepts** (BOC) by mapping words to their related unambiguous concepts by means of semantic resources (Bloehdorn *et al.*, 2006). This mapping is known by *conceptualization*. The resulting representation can also be enriched using other related concepts that may help in classification.

Authors in (Bloehdorn *et al.*, 2006) proposed using BOC for text representation and tested it through experiments using AdaBoost algorithm on three different corpora. During experiments on the corpus Reuters-21578, WordNet is used as the background knowledge whereas MeSH medical ontology is used with Ohsumed dataset. As for the third corpus FAODOC, the ontology AGROVOC was used as semantic resource. Different strategies of sense disambiguation were used. Moreover, the superconcepts of specific concepts, that are discovered in text are also integrated into the vector of concepts for representing text documents. These superconcepts are searched up to a maximal distance into the ontology. This process is known by *generalization* which, according to authors, improved classification results when applied to the general purpose background knowledge WordNet. Authors conclude that applying generalization in domain specific tasks where domain ontology is used for conceptualization is not adequate. In fact, adding more general concepts to text representation might introduce noise to the feature space and thus it disturbs classification.

Authors in (Bai *et al.*, 2010) choose a fully automated conservative method to align three general purpose semantic resources: WordNet, OpenCyc and SUMO and use the resulting knowledge base in conceptualization. By means of this knowledge base, the proposed system replaces the classical BOW model by BOC through semantic text indexing of documents. As for ambiguous words, the system chooses the most appropriate concept

according to the context of the word in order to choose the most appropriate meaning the word conveys. As for text classification, authors tested SVM on three different corpora: Reuters-21578 (Lewis *et al.*, 2004), Ohsumed (Hersh *et al.*, 1994) and 20Newsgroups (Rennie, 2013) where text is represented using the new BOC model. Authors reported significant improvements especially with Ohsumed data set. Authors concluded that semantic text representation is particularly effective for domain specific text classification (Bai *et al.*, 2010).

Authors in (Guisse *et al.*, 2009) propose also a BOC-based approach for patent classification using domain ontology. To involve superconcepts or more general concepts in text representation, authors propose a weight propagation algorithm that attributes appropriate weights to superconcepts in the ontology. In fact, after mapping patent text to a concept in the ontology, this algorithm weights the participation of its superconcepts in text representation, according to the distance between them and the mapped concept in the ontology (the number of links on the path linking them together throughout the hierarchy). This work demonstrated significant improvement in patent classification.

Authors in (Gabrilovich *et al.*, 2007) proposed *Explicit Semantic Analysis* (ESA) for text representation. In fact, authors enrich text representation with massive amounts of world knowledge by means of Wikipedia. First, authors build an inverted index on Wikipedia database of articles relating each word to the articles where they occur. When a text is treated in the proposed system, its words are mapped to articles using the previously built index. As each article can be considered as a concept, the treated text can be represented by a vector of concepts. Authors argue that their semantic interpretation methodology is capable of resolving ambiguities as they consider the neighbors of ambiguous words. This approach was evaluated in the context of word similarity on WordSimilarity-353 collection (Finkelstein *et al.*, 2002) and was also applied to document similarity on a dataset retrieved from the Australian Broadcasting Corporation's news mail service (Lee *et al.*, 2005). Authors reported an improved correlation with human judgment on relatedness in both tasks as compared with the traditional BOW, LSA and the same approach using ODP as the semantic resource.

Authors in (L. Huang *et al.*, 2012) extend the previous approach and use both Wikipedia and WordNet to find candidate concepts for text representation and also to enrich this representation with related concepts. In fact, this approach proposes a framework for learning document similarity using different features at different levels like cosine similarity at document level, relatedness between concepts at concept level and finally relatedness between groups of concepts at topic level. For example, concept vectors are enriched with related concepts that are weighted proportionally to their semantic similarity with the most similar concepts of the vector.

As compared with previous approach on the dataset retrieved from the Australian Broadcasting Corporation's news mail service, this approach showed a better correlation with human judgment. The learned similarity measure is then tested on four corpora derived from the well-known corpora: Reuters-21578 (Lewis *et al.*, 2004), Ohsumed (Hersh *et al.*, 1994) and 20Newsgroups (Rennie, 2013). Authors (L. Huang *et al.*, 2012) reported highest improvement in document classification using K-Nearest Neighbors KNN (Soucy *et al.*, 2001) and also in

document clustering using K-means (MacQueen, 1967) when applied on the medical dataset derived from Ohsumed (Hersh *et al.*, 1994).

In fact, using concepts as alternative features to words in text representation seems to be promising. State of the art approaches using a BOC for text representation demonstrated improvements in text classification, text clustering and in other IR tasks. The studied approaches covered many application domains using general purpose or domain specific semantic resources.

3.1.3.3 Comparison

This section overviewed some state of the art works all aiming to improve the classical BOW and overcome its limitation. The proposed extensions introduced alternative features to the original BOW extending the representation model. In fact, phrases or N-grams (Caropreso *et al.*, 2001; Z. Li *et al.*, 2009) and concepts (Bloehdorn *et al.*, 2006; Gabrilovich *et al.*, 2007; Guisse *et al.*, 2009; Bai *et al.*, 2010; L. Huang *et al.*, 2012) were the major candidates resulting in BOP and BOC respectively. Many authors reported significant improvements related to these alternative models in text classification (Caropreso *et al.*, 2001; Bloehdorn *et al.*, 2006; Guisse *et al.*, 2009; Z. Li *et al.*, 2009; Bai *et al.*, 2010; L. Huang *et al.*, 2012) and also in other tasks like clustering (L. Huang *et al.*, 2012), IR (Renard *et al.*, 2011; Dinh *et al.*, 2012), document similarity and word similarity (Gabrilovich *et al.*, 2007).

Table 12 compares both phrases and concepts as alternatives to words in the BOW according to four criteria:

- Good statistics: the capability of the representation model to capture language statistics
- Captures co-occurrences: the capability of the representation model to capture word co-occurrences or the words that usually occur together or close in text
- Captures semantics: the capability of the representation model to capture the meanings the words convey
- Captures context: the capability of the representation model to capture the context of the word as occurred in text and to take this context into consideration to choose the adequate feature

In fact, using words in the BOW is useful for collecting good statistics on text. Nevertheless, using only words in the models ignores words' co-occurrences and words' contexts which imply ambiguities in the model. On the contrary, phrases embed poor statistics on the text but they can capture word co-occurrences and contexts which help in resolving some ambiguities. Concepts seem to be the least ambiguous and the best compromise as compared with words and phrases.

In fact, most works consider *Concepts* as the best alternative to words since it encompasses the identified drawbacks of the classical BOW. First of all, concept replaces synonymous words that are related to its sense which overcomes redundancy. Second, concept has one explicit meaning which resolves ambiguities. Finally, relations between concepts can be measured and quantified according to the semantic resource in use such as thesauri or domain specific ontologies. These relations help involving related concepts in text representation through generalization or in prediction (see section 3.3).

To choose concepts as an alternative feature to words, they have to offer advantages more than words. This is true for capturing context, semantics and co-occurrences but not for statistics where words provide higher statistical quality. A combination of concepts and words in a hybrid representation model seems to be another option as words fill the gaps in concepts and vice versa.

Feature	Good statistics	Captures co-occurrences	Captures semantics	Captures context
Word	+++	-	-	-
Phrase	+	++	+	++
Concept	++	+	++	+++

Table 12. Comparing alternative features of the VSM. (+, ++, +++): degrees of support (-): unsupported criterion

3.1.4 Discussion

This section surveyed the state of the art approaches that implicate the semantics, as discovered in text, in the representation model extending the classical BOW model. Using either implicit or explicit semantics, most of their results demonstrated improvements in classification and other tasks related to the IR domain as well. The three previous approaches are compared in Table 13.

Latent Topic modeling looks for statistically related groups of terms by observing term co-occurrences in an input collection. Resulting groups, so called latent topics, are highly dependent on the initial collection; they cannot be generalized to cover unseen terms of new documents. Furthermore, topic modeling does not provide explicit semantic interpretations on latent topics or the relations among them (J. Z. Wang *et al.*, 2007; L. Huang *et al.*, 2012). Finally, these methods are unsupervised and their adaptation to supervised text classification is very expensive (Aggarwal *et al.*, 2012).

Semantic Kernels are usually used with SVM classifiers to project text representations to a new space where finding a classification model is easier with a good class prediction capability. Semantic kernels use either implicit or explicit semantics and seem to help SVM in classification when compared with other kernels according to the literature (Séaghdha *et al.*, 2008; Wang *et al.*, 2008; Séaghdha, 2009).

Alternative features for the VSM were also investigated. Some works used phrases to replace words in a new BOP representation while others chose concepts instead for a BOC model. After a comparative study on the three alternatives (see Table 12), *Concepts* can be considered as the best alternative to words and thus the BOC is the best extension to the classical BOW. In fact, concepts proved to be the best compromise that conveys good statistics and resolves ambiguities overcoming many limitations of the classical BOW like redundancy and ambiguities. Furthermore, relations between concepts can be measured and quantified according to the deployed semantic resource such as thesauri or domain specific ontologies. These relations help involving related concepts in text representation through generalization (Bloehdorn *et al.*, 2006) and vector enrichment (L. Huang *et al.*, 2012).

Approach	Basic principle	Advantages	Disadvantages
Latent topic modeling	Term co-occurrences in text convey meaning	Discover implicit concepts in text	Unsupervised and needs adaptation for supervised classification
Semantic kernels	Project text representation into another feature space	Transform the training set to a linearly separable set	Deployed with SVM only Uses topic modeling, alternative features or other methods for projection
Alternative features	Use phrases or concept instead of words	Represent text with explicit semantics	Requires semantic resources and NLP

Table 13. Comparing latent topic modeling, semantic kernels and alternative features for integration semantics in text indexing

Many authors reported significant classification improvements using concepts as an alternative feature or a hybrid model combining words and concepts. Furthermore, results proved that conceptualized representation is particularly beneficial in classifying domain specific text where different classifiers showed difficulty in class prediction as reported in our experimental study (see chapter 2). In addition, relations and similarities between concepts are explicitly expressed in semantic resources and can be measured and used in enriching concept based text representation, this can also lead to an effective class prediction. Thus, we choose the use of concepts as an alternative feature to involve explicit semantics in text representation or in prediction aiming to improve classification performance.

3.2 Involving semantics in training

To involve semantic features in training (Figure 21, arrow 2), this family of approaches uses ontologies as a basis for classification; classification model is either the entire ontology or part(s) of its hierarchy. In these approaches, concepts replace words in text representation. In addition, the hierarchy and the relations among the added concepts are taken into consideration in the training phase which affects the learned model.

We introduced the notion of *generalization* in a former section. Both works (Hotho *et al.*, 2003; Guisse *et al.*, 2009) used the hierarchical structure of semantic resources to involve related concepts in text representation. Authors in (Guisse *et al.*, 2009) used propagation algorithm to propagate the weights of identified concepts in patents to their superconcepts. Furthermore, authors in (L. Huang *et al.*, 2012) used similar concepts in order to enriched text representation and proposed the approach *Enriching vectors*. Similarities among concepts are assessed using relations between concepts in the semantic resource. Both *Generalization* (Hotho *et al.*, 2003; Guisse *et al.*, 2009) and *Enriching vectors* (L. Huang *et al.*, 2012) involve related semantics into text representation which involves semantics in the classification model *implicitly*. The following subsections introduce two approaches that involve *explicitly* the hierarchy of ontology in text representation and training as a classification model: *Semantic Trees* and *Concept Forests*. Then the discussion compares both explicit and implicit approaches for involving semantics in building the classification model during training.

3.2.1 Semantic trees

Semantic trees (Peng *et al.*, 2005) are hierarchies where each node or concept is assigned an importance score according to its observed occurrences in the training dataset for each category. Figure 22 illustrates the steps of text representation. First, words are mapped to WordNet synsets and their weights are attributed to the corresponding synset. When two or more words are mapped to the same synset, their weights are accumulated. Finally, the weights are normalized and propagated throughout the hierarchy resulting in a weighted WordNet for each category. These semantic trees constitute the classification model learned during the training phase.

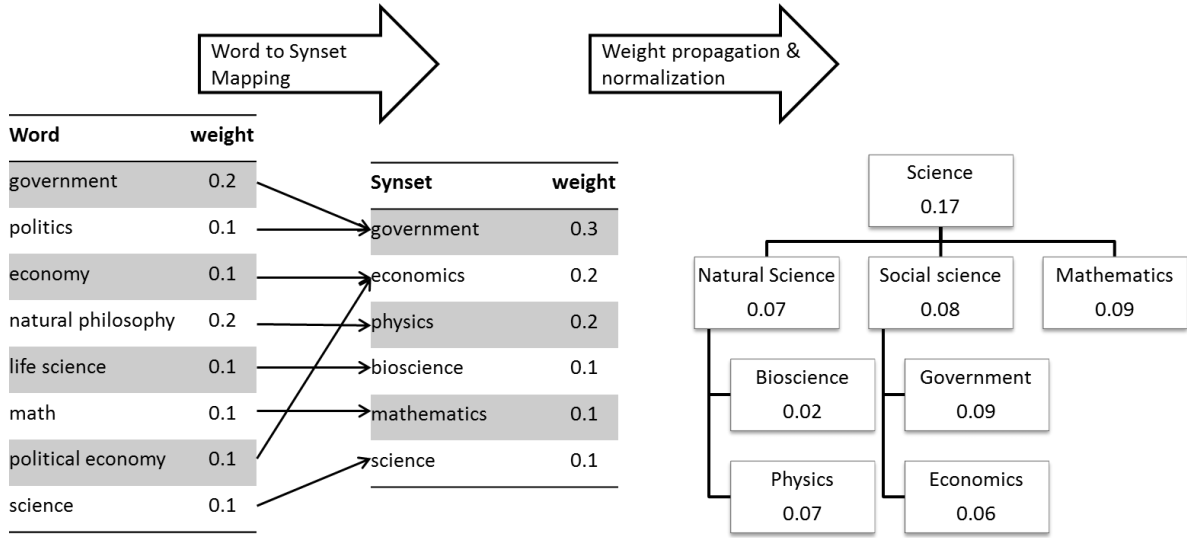


Figure 22. Mapping words that occurred in text to their corresponding synsets in WordNet and accumulating their weights when multiple words are mapped to the same synset like government and politics. Then, accumulated weights are normalized and propagated on the hierarchy (Peng *et al.*, 2005)

To predict the class of a new document, its text is represented as a weighted semantic tree as well and thus classification is a method that can compare it with the semantic tree of each category and finally the document is assigned to the most similar category. Authors proposed a similarity measure that is inspired from the classical cosine measure and reported significant improvement in classifying Yahoo! documents. Given a document d and a category c_k the similarity is assessed using the following formula:

$$Sim(d, c_k) = \frac{\sum_{i=1}^n (d_i * c_{k,i})}{\sqrt{\sum_{i=1}^n d_i^2} * \sqrt{\sum_{i=1}^n c_{k,i}^2}} \quad (27)$$

Where:

n is the number of concepts in the hierarchy

d_i is the weight of the concept i in document representation

$c_{k,i}$ is the weight of the concept i in the representation of the category k

3.2.2 Concept Forests

Concept Forests (J. Z. Wang *et al.*, 2007) are parts of the hierarchy of WordNet. Authors constituted these forests by mapping words found in text to the synsets of WordNet taking into account their context in text to identify the meaning they convey. Authors deployed purification algorithm to clear noisy concepts that can affect prediction capabilities. These forests are then used as a text representation model and tested on Reuters-21578 for text clustering using K-means. Authors reported performance improvement using the new approach.

Having text documents represented as concept forests similarly to the example of text representation using parts of WordNet in Figure 23, authors proposed a simple similarity measure to compare documents using the sets of synsets by which they are represented. This similarity is given as follows:

$$Sim(d_1, d_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} \quad (28)$$

Where:

S_1, S_2 are sets of synsets representing documents d_1, d_2 respectively.

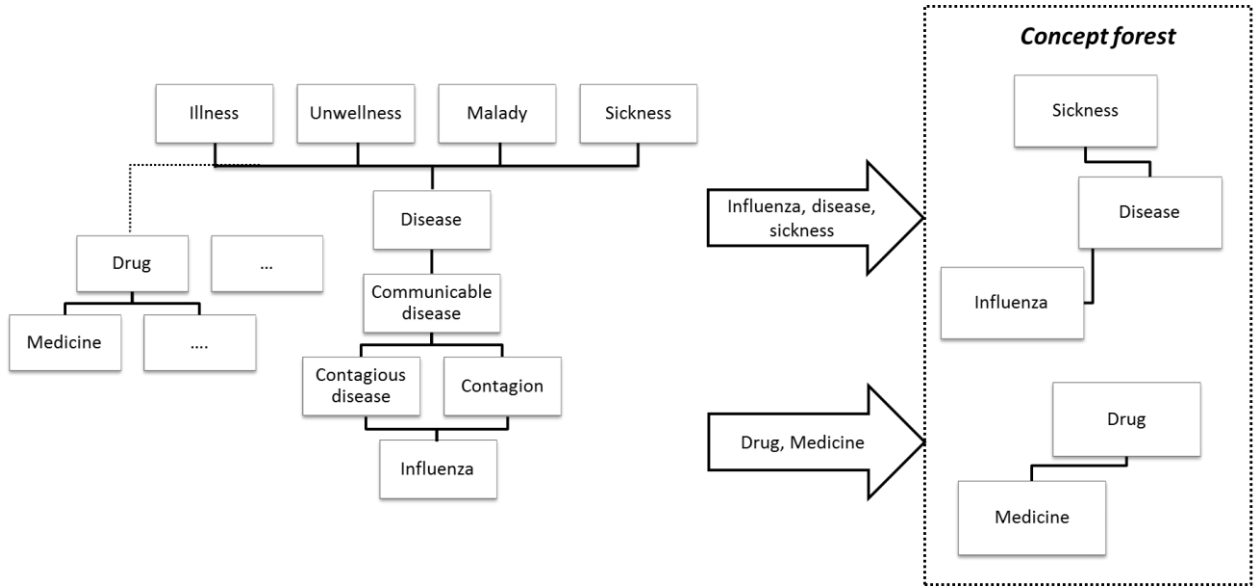


Figure 23. Building a concept forest for a text document that contains the words: "Influenza", "Disease", "Sickness", "Drug", "Medicine" (J. Z. Wang *et al.*, 2007).

3.2.3 Discussion

The previous sections presented two approaches proposed in the state of the art involving explicitly the semantic hierarchy in training. The first approach uses the whole hierarchy (Peng *et al.*, 2005) while the second one uses parts of it (J. Z. Wang *et al.*, 2007) as a classification model. These models demonstrated certain effectiveness when applied to text classification.

Table 14 compares both approaches with implicit approaches: generalization and enriching vectors. Though semantic trees and concept forests approaches showed promising results, their major drawback is the intensive use of semantic resources that can affect the efficiency of text classification. In fact, semantic trees (Peng *et al.*, 2005) use weight propagation algorithm to propagate weights from synsets related directly to text to other synsets in the hierarchy. Moreover, using all synsets of WordNet might introduce noise into the system and disturb classification.

Concept forests (J. Z. Wang *et al.*, 2007) use parts of WordNet and eliminate noisy synsets from the forest by means of purification nevertheless the similarity measure proposed by the authors is a simple formula that counts the number of common concepts between two forests.

Approach	Basic principle	Application	Advantages	Disadvantages
Generalization (Hotho <i>et al.</i> , 2003; Guisse <i>et al.</i> , 2009)	-Incorporating subsuming concepts in vectors	Clustering	-Enrich text representation with more general concepts	-Inadequate when using MeSH -Requires adequate formulas to attribute weights to added concepts
Enriching vectors (L. Huang <i>et al.</i> , 2012)	-Incorporating related concepts in vectors using WordNet and Wikipedia	Clustering and Classification	-Enrich text representation with related concepts (generalization is a special case)	-Requires adequate similarity measures on the ontology to attribute weights to added concepts
Semantic trees (Peng <i>et al.</i> , 2005)	-Uses WordNet as a model with importance weights	Classification	-Involves all the ontology in text representation	-Requires weight propagation techniques -Adds noise to text representation
Concept forests (J. Z. Wang <i>et al.</i> , 2007)	-Construct forests of semantic trees using synsets from WordNet	Clustering	-Involves relevant parts of the ontology in text representation -Limits noise using purification	-Requires weight propagation techniques -Similarity between vectors is based on commonalities between their representing concept sets

Table 14. Comparing Generalization, Enriching vectors, Semantic trees and Concept forests in involving semantics in training

As for generalization (Hotho *et al.*, 2003; Guisse *et al.*, 2009) and Enriching vectors (L. Huang *et al.*, 2012), both approaches keep the original vector as a representation model of text and incorporates related concepts to those detected in text to enrich this representation. The number of added concepts to the vector can be limited in order to avoid adding noise to the feature space and adequate weighting formulas are also required to attribute weights to the added concepts. Generalization enriches vectors with the subsumers of their concepts whereas enriching vectors enriches vectors with the concepts that are related to them with an IS-A

relation (like generalization) or any other semantic relation. In other words, generalization can be considered as a special case of enriching vectors.

In fact, involving semantic resources in representation has influences on the classification model that a supervised classifier learns during training phase and on class prediction as well. This will be discussed in next section.

3.3 Involving semantics in class prediction

According to the literature, most research focused on enriching text representation with semantics and used classical techniques for prediction; for example, authors in (Peng *et al.*, 2005; Gabrilovich *et al.*, 2009) used the classical Cosine similarity measure to assess text-to-text similarity. Only few works tried to involve semantics in class prediction (Figure 21, arrow 3) by proposing new *Semantic Text-To-Text Similarity Measures*.

Concept Forests, proposed in (J. Z. Wang *et al.*, 2007), are parts of the hierarchy of WordNet that are composed of the synsets related to the treated text. Authors used these forests as the text representation model and as the classification model as well. As for assessing the similarity between two documents, authors chose to use a relatively simple formula to compare the concept forests representing their text. This formula is an adapted version of the Jaccard similarity measure introduced earlier in chapter 2 and applied to the sets of terms representing both documents according to the formula (28). Authors validated this similarity measure on small corpora derived from the corpus Reuters-21578 and demonstrated improvement in text classification. In this approach, the participation of semantics in prediction is limited; it takes into consideration the commonalities between the sets of concepts ignoring the potential similarities among the concepts of these sets.

New semantic approaches for assessing text-to-text similarities seem to be feasible using semantic similarities among concepts pair-to-pair. In fact, such approaches involve semantics in document comparison and in class prediction as well by discovering similarities between texts considering semantically similar terms in addition to lexically similar ones. According to the literature, assessing the semantic similarity between concepts of semantic resources attracted the attention of many researchers which resulted in proposing numerous semantic similarity measures. In fact, each of these measures pretends to have the maximum correlation with human judgments when assessing similarities among concepts (Al-Mubaid *et al.*, 2006; Pirro, 2009; Sanchez *et al.*, 2012). We will present some of these measures in details in section 4.

As presented earlier, authors in (Guisse *et al.*, 2009) proposed a propagation algorithm to attribute weights to subsumers involving them in text representation. Furthermore, authors proposed a new text-to-text similarity measure based on these weights as well as the semantic similarity between concepts pair-to-pair. This new similarity measure is the prediction criterion that replaces classical text-to-text similarity measures introduced earlier in chapter 2. Authors reported better clustering of patents using semantic similarities (Guisse *et al.*, 2009). The similarity measure is given by the following formula:

$$SemSim(P_1, P_2) = \sum_{c_i \in C^1, c_j \in C^2} \frac{|sim(c_i, c_j)|}{1 + |P_{c_i}^{P_1} - P_{c_j}^{P_2}|} \quad (29)$$

Where:

P_1, P_2 are the patents to compare

C^1, C^2 are the groups of concept that represent P_1, P_2 respectively

$|sim(c_i, c_j)|$ is the semantic similarity between two concepts that is estimated to the normalized distance between the concepts in the domain ontology.

$P_{c_i}^{P_1}$ is the weight of concept c_i in patent P_1 , the weight issued from text statistics and the application of weight propagation algorithm.

The main difference between this approach and the one proposed in (J. Z. Wang *et al.*, 2007) is that the text-to-text similarity formula aggregates the semantic similarities among concepts pair-to-pair into a semantic similarity among two groups of concepts. In other words, this approach involves semantics not only in text representation and classification model but also in assessing text-to-text similarity and in class prediction taking into considerations that patents are represented by groups of concepts.

Similarly to the previous approach, many authors proposed aggregation functions for text-to-text similarity. Most of them used an average of the semantic similarities among concepts used in text representation pair-to-pair (Rada *et al.*, 1989; Azuaje *et al.*, 2005; Hao *et al.*, 2008). Others preferred more sophisticated functions that we survey next.

Authors in (Hliaoutakis *et al.*, 2006) aggregated the semantic similarity between concepts of a queries and document in the following formula and tested it on MEDLINE documents using MeSH ontology for a semantic IR in the medical domain.

$$SemSim(q, d) = \frac{\sum_i \sum_j q_i d_j sim(c_i, c_j)}{\sum_i \sum_j q_i d_j} \quad (30)$$

Where:

$q_i d_j$ are the weights of concept i in the query and the concept j in a document

$sim(c_i, c_j)$ is the similarity between the concept c_i from the query q and c_j from the document d .

The previous formula is an adapted version of the well-known cosine similarity integrating semantic similarity between concepts of the query and the document pair-to-pair. This involves semantic similarity in document ranking. the Authors reported improved precision and recall in IR on MEDLINE documents.

Authors in (Mihalcea *et al.*, 2006; Mohler *et al.*, 2009) developed a different aggregation function for comparing short texts or phrases. In fact, they compare each concept from a text with all concepts of the other text to identify the maximum similarity. The aggregation function is the average of resulting similarities weighted using the Inverse Document Frequency (Idf) of the treated concepts following this formula:

$$SemSim(T_1, T_2) = \frac{1}{2} \left(\frac{\sum_{w \in T_1} maxSim(w, T_2) * idf(w)}{\sum_{w \in T_1} idf(w)} + \frac{\sum_{w \in T_2} maxSim(w, T_1) * idf(w)}{\sum_{w \in T_2} idf(w)} \right) \quad (31)$$

Where:

$\maxSim(w, T_2)$ is the maximum similarity between word (w) and all words in (T_2)

This function improved significantly text-to-text similarity on Microsoft paraphrase corpus (Dolan *et al.*, 2004) as compared to the classical Cosine similarity measure (Mihalcea *et al.*, 2006). It demonstrated high accuracy when applied to automatic short answer grading (Mohler *et al.*, 2009). The main drawback is that this approach ignores all dependencies between words in sentences.

Authors in (L. Huang *et al.*, 2012) developed a supervised approach for combining multiple semantic features in a semantic similarity function with the maximum correlation with human judgments using both WordNet and Wikipedia as semantic resources. Among the combined features we mention particularly the classical cosine measure applied to enriched vectors. In fact, vector enrichment makes the compared vectors less sparse by enriching each vector with the concepts found in the other vector. Having two documents d_1, d_2 , the concept c that is detected in d_2 is missing in d_1 . To enrich d_1 with this concept, authors propose to assign it the following weight:

$$w(c, d_1) = w(SC(c, d_1)) * sim(c, SC(c, d_1)) * CC(c, d_1) \quad (32)$$

Where:

$w(SC(c, d_1))$ is the weight of the Strongest Connection of the concept c in d_1 which is the weight of the most similar concept in d_1

$sim(c, SC(c, d_1))$ is the similarity between the concept and its strongest connection

$CC(c, d_1)$ is the context centrality of the concept c in the document d_1

$$CC(c, d_1) = \frac{\sum_{c_i \in d_1} sim(c, c_i) * w(c_i, d_1)}{\sum_{c_i \in d_1} w(c_i, d_1)} \quad (33)$$

Where:

$sim(c, c_i)$ is the similarity between the concept c and the concept c_i from the document d_1 .

$w(c_i, d_1)$ is the weight of concept c_i in the document d_1 .

This approach uses semantic similarities among concepts to estimate the weight to use in enriching vectors. Moreover, authors propose supervised approach to learn a semantic measure to assess the similarity between documents using semantic features at three levels: concepts, groups of concepts and document level. This approach is the most developed and disciplined approach in the literature and demonstrated promising results when used in clustering and classification of small corpora derived from Reuters-21578 (Lewis *et al.*, 2004) and Ohsumed (Hersh *et al.*, 1994). Nevertheless, thorough test is needed in order to validate this approach on the entire corpora in order to prove its effectiveness and efficiency on large datasets.

In fact, few works proposed semantic measures for text-text similarity aggregating semantic similarities between concepts pair-to-pair. Some of them are tested for text classification while others for IR or short text classification. Most approaches propose functions based on an

average formula taking into consideration the size of the feature space or the weighting scheme used for text representation. Moreover, they were tested on relatively small corpora and in particular contexts. Accordingly, evidences on their effectiveness in text classification are still insufficient.

3.4 Discussion

This section presented a survey on the literature on works involving semantics in text classification. We identified three levels for integrating semantics: indexing, training and prediction. Then we presented some works that investigated the effect of this integration on different tasks related to the information retrieval domain. Different state of the art works are synthesized in Table 15.

According to the literature, most works investigated the effect of semantics on text treatment at the representation level after indexing (Caropreso *et al.*, 2001; Liu *et al.*, 2004; Bloehdorn *et al.*, 2007; Séaghdha *et al.*, 2008; Wang *et al.*, 2008; Aseervatham *et al.*, 2009; Z. Li *et al.*, 2009; Séaghdha, 2009). Some of these works deployed either implicit semantics using *latent topic modeling* (Liu *et al.*, 2004) that is inherently unsupervised and its adaptations to supervised classification is quite expensive (Aggarwal *et al.*, 2012). Others deployed explicit semantics using alternative features that transformed the classical BOW into *BOP* (Caropreso *et al.*, 2001; Z. Li *et al.*, 2009), or *BOC* (Bloehdorn *et al.*, 2006; Hliaoutakis *et al.*, 2006; Mihalcea *et al.*, 2006; Gabrilovich *et al.*, 2007; Guisse *et al.*, 2009; Bai *et al.*, 2010; L. Huang *et al.*, 2012) where phrases and concepts respectively cover textual features that words are not able to cover thus resulting models overcome the limitations of the classical BOW model: redundancy, ambiguity and orthogonality. Results of tests deploying explicit semantics demonstrated improvements in classification and other tasks related to the IR domain as well.

In addition to concepts which are considered the best alternative features, some works deployed relations among concepts in semantic resources in order to enriched text representation using *semantic kernels* with SVM classifiers (Bloehdorn *et al.*, 2007; Wang *et al.*, 2008; Aseervatham *et al.*, 2009; Séaghdha, 2009) while others used *generalization* in order to implicate superconcepts in text representation (Bloehdorn *et al.*, 2006). Authors in (L. Huang *et al.*, 2012) proposed a method to enrich compared documents mutually using semantic similarities among their concepts.

Enriching text representation using either of the preceding methods has most likely an impact on the training process of the classification technique. More intensive use of semantics in training is reported in (Peng *et al.*, 2005; J. Z. Wang *et al.*, 2007; Guisse *et al.*, 2009) where the semantic resource is used as a classification model after assigning each of its concepts a weight corresponding to its importance in the corpus. The major drawback of these approaches is that the intensive use of semantic resources can affect the efficiency of text classification. Thus, enriching text representation using similar concepts is more advantageous.

To the best of our knowledge, few works proposed semantic approaches that involve semantics in class prediction. Most approaches studied in this chapter developed similarity functions that aggregate the semantic similarity between concepts pair-to-pair in order to assess the similarity between two groups of concepts. These groups represented two texts (Mihalcea *et*

al., 2006), a class model and a document (Guisse *et al.*, 2009), a query and a document (Hliaoutakis *et al.*, 2006) or two documents (L. Huang *et al.*, 2012). Moreover, these approaches were developed in an ad hoc manner and tested on relatively small corpora (L. Huang *et al.*, 2012).

According to the literature, Authors seem to disagree on the utility of semantics in classification (Stein *et al.*, 2006). Nevertheless, it seems to be promising to take the application domain into consideration when developing a system for semantic classification (Ferretti *et al.*, 2008) or in other IR tasks (Renard *et al.*, 2011; Dinh *et al.*, 2012). For example, authors in (Bai *et al.*, 2010) reported interesting results when they applied their approach on the medical corpus Ohsumed. In addition, generalization or adding superconcepts to the feature space was inefficient when using the medical MeSH ontology (Bloehdorn *et al.*, 2006). Thus, it is essential to investigate and identify new approaches involving concepts and their semantic relations in different steps of the classification process and also to validate these approaches on large datasets. This affirms the intent of this work to develop enhanced semantic text classification that can meet human judgment and the choice of the medical domain as an application domain for this work.

Next section presents a state of the art of semantic similarity measures usually used in assessing the similarity between two concepts in an ontology. This semantic similarity was mentioned earlier in this section as it is widely used in enriching text representation and also in assessing text-to-text semantic similarity.

Reference	Involve semantics in			Basic principle	Semantic resource	Dataset	Task	Advantages	Disadvantages
	Indexing	Training	Prediction						
Liu <i>et al.</i> (2004)	X			-Latent topic modeling -Supervised LSI		-Reuters-21578 -Industry Sector	Text classification using SVM	-Identify the discriminative features of each of the classes with local LSI	-No use of resource is used for explicit semantics -Difficult to compare documents across subspaces -Computationally expensive
Séaghdha <i>et al.</i> (2008)	X			-Distributional kernels -Probability of co-occurrences		SemEval 2007	Classification of semantic relations between nominal with SVM	-Distributional kernels are more effective than Classical kernels	-No use of resource is used for explicit semantics
Séaghdha (2009)	X			-Semantic kernel	WordNet	SemEval 2007	Classification of semantic relations between nominal with SVM	Higher level of performance as compared to other systems in SemEval 2007	-Approach specific for WordNet -uses only hierarchical relations (Hyponymy, hypernymy)
Bloehdorn <i>et al.</i> (2007)	X			-Semantic and syntactic kernels	WordNet	TREC 8, 9, 10	Question Classification using SVM	-Semantic similarity on WordNet for semantic kernel -Language structure for syntactic kernel	-Approach specific for WordNet
Wang <i>et al.</i> (2008)	X			-Semantic kernel	Wikipedia	-Reuters-21578 -Ohsumed -20NewsGroups -Movies	Text classification using SVM	-Using semantic similarities, similar concepts can be added making vector less sparse	-Using heuristics to limit the number of the most similar concepts used in enriching vectors
Aseervatham <i>et al.</i> (2009)	X			-Semantic kernel	UMLS	-2007 CMC Medical NLP International Challenge	Semi structured text classification	-Used UMLS and semantic similarity to constitute domain specific semantic kernel	-Doesn't resolve ambiguities
Caropreso <i>et al.</i> (2001)	X			-BOP		Reuters-21578	SVM and Rocchio	-Takes into account frequently occurring bigrams during indexing -Improves classification effectiveness	-No use of resource is used for explicit semantics -Excessive use of bigrams causes deterioration in effectiveness -Sparseness and ambiguities
Z. Li <i>et al.</i> (2009)	X			-BOP		From ACM digital library	KNN, Decision Trees, SVM and NB	-Takes into account frequently occurring N-grams during indexing -Improves classification effectiveness on focused datasets	-No use of resource is used for explicit semantics -Sparseness and ambiguities in feature space
Bloehdorn <i>et al.</i> (2006)	X	X		-BOC with generalization and disambiguation	-WordNet -MeSH -AGROVOC	-Reuters-21578 -Ohsumed -FAODOC	Classification using AdaBoost	-Uses explicit unambiguous conceptual knowledge in text representation -Implicate the superconcepts	-Generalization deteriorates effectiveness with domain specific ontology

Reference	Involve semantics in			Basic principle	Semantic resource	Dataset	Task	Advantages	Disadvantages
	Indexing	Training	Prediction						
Bai <i>et al.</i> (2010)	X			-BOC with disambiguation	-WordNet -OpenCyc -SUMO	-Reuters-21578 -Ohsumed -20Newsgroups	Text classification using SVM	-Significant improvement especially with Ohsumed	-uses a conservative fully automated algorithm for aligning the ontologies
Gabrilovich <i>et al.</i> (2007)	X			-BOC with disambiguation	-Wikipedia	-WordSimilarity-353 -Australian Broadcasting Corporation's news mail service	-Word similarity -Document similarity	-Improved correlation with human judgment on relatedness as compared with BOW, LSA and the same approach using ODP as the semantic resource.	-Wikipedia doesn't cover specific domains
Peng <i>et al.</i> (2005)	X	X		-Semantic trees for representation and class model	-WordNet	-Yahoo! documents	-Classification using Cosine similarity	-Weight propagation implicates all synsets of WordNet in representation and in class model	-Requires weight propagation techniques -Adds noise to text representation
J. Z. Wang <i>et al.</i> (2007)	X	X	X	-Forests of semantic trees	-WordNet	-Reuters-21578	-Clustering using K-means	-Involves relevant parts of the ontology in text representation -Limits noise using purification	-Requires weight propagation techniques -Similarity between forests=number of common concepts
Guisse <i>et al.</i> (2009)	X	X	X	-BOC with weight propagation	-Patent ontology	-Patents	-Clustering	-Uses semantic distance when comparing patents -Implicate the superconcepts in representation -Uniform clusters distribution	-Requires weight propagation techniques -Requires consistent finely grained ontology
Hliaoutakis <i>et al.</i> (2006)	X		X	-BOC & adapted Cosine using semantic similarities	-MeSH	-MEDLINE documents	-IR	-adapted version of Cosine using semantic similarities between concepts in query and document	-Requires parameter tuning
Mihalcea <i>et al.</i> (2006)	X		X	-BOC with semantic similarity	-WordNet	Microsoft paraphrase corpus	-Text to text similarity -Automatic short answer grading	-New text to text semantic similarity using Idf and semantic similarity between concepts P2P	-Ignores all dependencies between words in sentences
L. Huang <i>et al.</i> (2012)	X	X	X	-BOC with semantic similarity	-Wikipedia -WordNet	-Reuters-21578 -Ohsumed	-Clustering using K-means -Classification using KNN	-Enriches compared documents mutually with missing concepts -uses semantic similarity at concept, groups of concepts and document level	-Very complex approach -Tested on small corpora

Table 15 Involving semantics in text representation comparison and in learning class model

4 Semantic similarity measures

Computing semantic similarity between concepts has been an important issue to many research domains as linguistics, artificial intelligence bio-medicine, IR, ontology alignment, and knowledge-based systems. According to authors in (Petrakis *et al.*, 2006) “*Semantic Similarity relates to computing the similarity between concepts (terms) which are not necessarily lexically similar*”. Existing metrics estimate the semantic similarity (common shared information) between two concepts according to certain language or domain resources like terminologies, corpora, etc.

Let O be the ontology with IS-A hierarchical links and other semantic relations and let $(c_1, c_2) \in O$ be a pair of concepts from the ontology. Next paragraphs present some of the proposed measures in the literature where each measure tries to estimate the similarity between them according to a particular hypothesis.

Reviewing the literature, authors suggested different categorization schemes for these metrics. So they can be organized into various and not necessarily disjoint categories (Pirro, 2009; Sanchez, Batet, *et al.*, 2011). We distinguished three major families of semantic similarity measures: Ontology-based measures, Information theoretic-based measures and Feature-based measures. In addition, hybrid measures combine multiple principles from different families. The first four subsections present the preceding families respectively which are compared in the fifth subsection.

4.1 Ontology-based measures

Measures belonging to this family are based on *the theory of spreading activation* (Cohen *et al.*, 1987; G. Salton *et al.*, 1988). One of its assumptions is that the hierarchy of concepts is organized along the lines of semantic similarity, so the meaning of a concept is highly related to the associated concepts (Cohen *et al.*, 1987). Thus, the closer the concepts the more similar they are (Hliaoutakis, 2005).

These measures are also called *path-finding measures* or *structure-based measures*. They depend only on the structure of the ontology in order to estimate the similarity between two of its concepts. Some of these measures depend only on the length of the path between concepts they are so called “*Path-based Measures*” while others take into consideration the position of concepts as well, so they are called “*Path and Depth-based Measures*”.

4.1.1 Path-based similarity measures

These measures estimate the similarity between two concepts using the number of the taxonomic links (IS-A) relating them in the ontology. Nevertheless, they ignore all other knowledge or information represented in the ontology such as the position of the concepts or the relations between them and other related concepts.

In *Rada et al* (Rada *et al.*, 1989) the similarity between two concepts is the length of the shortest path between them. The length of a path is the number of edges on this path which is given by the following formula:

$$Sim_{Rada}(c_1, c_2) = \min_{i \in [1, N]} |path_i(c_1, c_2)| \quad (34)$$

Where:

$path_i$ is the number of nodes between c_1 and c_2

i is in the range $[1, N]$

N is the number of possible paths between these concepts in the ontology.

This similarity measure estimates how close in the ontology the two compared concepts are. In fact, this efficient measure is the simplest one in this category; most of the rest is based on the same simple hypothesis with some variations. Through these variations, next measures take into consideration other factors in addition to the shortest path in order to improve the performance of the original measure (Hliaoutakis, 2005). For example, in Figure 24, $sim('oral\ thrush', 'athlete's\ foot') = 3$. This measure is adapted to the UMLS ontology in (Caviedes *et al.*, 2004).

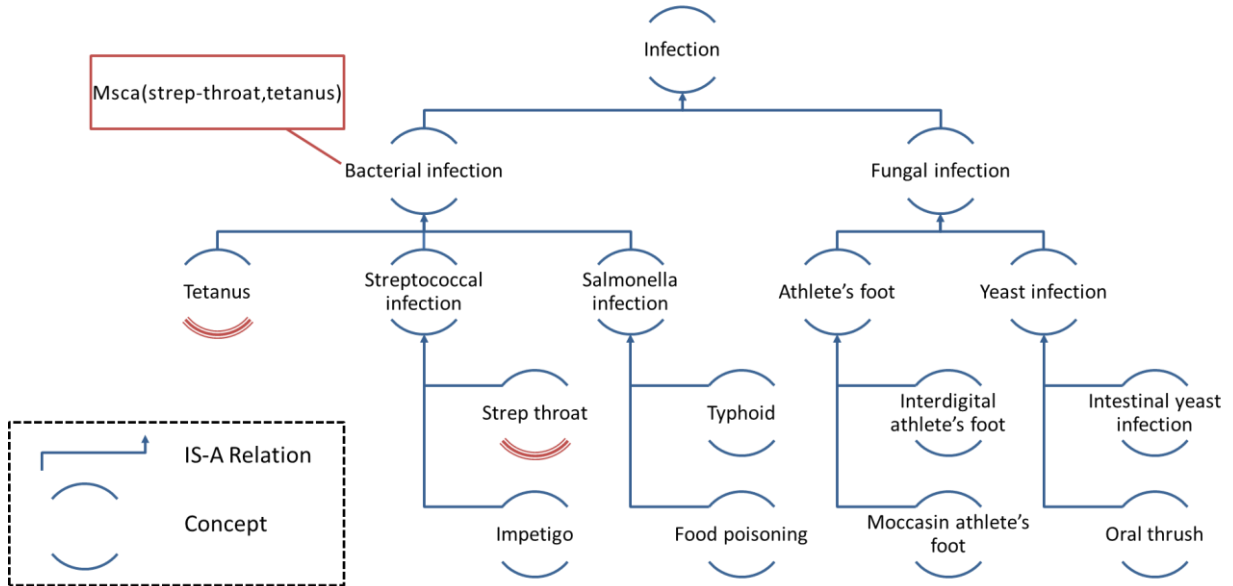


Figure 24. A part of UMLS (Pedersen *et al.*, 2012). The concept “bacterial infection” is the Most Specific Common Abstraction (msca) of “tetanus” and “strep throat”.

In *Bulskov et al* (Bulskov *et al.*, 2002), the authors propose an improved measure for estimating the similarity between two concepts as in the following formula:

$$Sim_{sp}(c_1, c_2) = 2 * MAX - Sim_{Rada}(c_1, c_2) \quad (35)$$

Where:

MAX is the longest path between two concepts in the ontology.

For the same previous example in Figure 24, $sim('oral\ thrush', 'athlete's\ foot') = 2 * 6 - 3 = 9$. The longest path's length equals to 7 (between ‘oral thrush’ and ‘food poisoning’). This measure was applied to query evaluation and document ranking.

4.1.2 Path and depth-based similarity measures

These measures estimate the similarity between two concepts using the number of the taxonomic links (IS-A) relating them in addition to their position or depth in the ontology and the other related concepts like their *Most Specific Common Abstraction* (msca). Taking depth into consideration in similarity calculations is based on the hypothesis that paths between deeper concepts in the hierarchy travel less semantic distance.

In **Wu et al** (Wu et al., 1994), the authors introduce a new concept to the hypothesis. The position of the most specific common concept c is considered in this measure. This concept is the closest common parent that is connected with the least number of IS-A links to the concepts (c_1, c_2) after taking all possible paths into account. The proposed measure in (Wu et al., 1994) is given by the following formula:

$$Sim_{w\&p}(c_1, c_2) = \frac{2 * depth(c)}{depth(c_1) + depth(c_2)} = \frac{2H}{N1 + N2 + 2H} \quad (36)$$

Where:

$N1$ and $N2$ are the the number of IS-A links connecting the most specific common concept to c_1 and c_2 respectively.

H is the number of IS-A links between c and the root of the ontology.

For example, according to Figure 24:

$$Sim('oral\ thrush', 'athlete's\ foot') = 2 * 1 / (2 + 1 + 2 * 1) = 0.4.$$

This measure was originally proposed to assess semantic similarity between verbs and also et estimate the effect of lexical choice on Chinese to English *Machine Translation* (MT). According to this measure, semantic similarity among two concepts has a score that varies between 0 and 1.

In **Leacock et al** (Leacock et al., 1998), the authors combine the shortest path between the compared concepts (using node counting) (Rada et al., 1989) with the maximum depth of the ontology according to the following formula:

$$Sim_{lch}(c_1, c_2) = \text{Max} \left[-\log \left(\frac{Sim_{Rad}}{2D} \right) \right] \quad (37)$$

Where:

D is the maximum depth of the ontology.

In Figure 24, $Sim('oral\ thrush', 'athlete's\ foot') = -\log(4/2 * 3) = 0.176$. In the cases where the ontology is composed of multiple subtrees with no common root node, D is the maximum depth of the subtree that contains the most specific common abstraction (Msca) or the lowest common subsume (LCS).

In *Li et al* (Y. Li *et al.*, 2003), the authors derive a non-linear function to estimate the semantic similarity between two concepts in WordNet. The intuition behind this function is that in order to calculate the similarity in a finite interval on infinite knowledge sources, a non-linear function is necessary. This function combines the shortest path measure (Rada *et al.*, 1989) along with the depth of the most specific common concept H .

$$Sim_{Li}(c_1, c_2) = e^{-\alpha \cdot Sim_{Rada}} * \frac{e^{\beta H} - e^{-\beta H}}{e^{\beta H} + e^{-\beta H}} \quad (38)$$

Where:

$\alpha \geq 0$ and $\beta > 0$ are parameters to configure.

According to the authors in (Y. Li *et al.*, 2003), optimal values of α and β are 0.2 and 0.6 respectively and the resulting scores are between 0 and 1. Given the same concepts as the preceding measures (see Figure 24), the similarity between them is estimated according to the following formula:

$$Sim('oral\ thrush', 'athlete's\ foot') = e^{-0.2*3} * \frac{(e^{0.6*1} - e^{-0.6*1})}{(e^{0.6*1} + e^{-0.6*1})} = 0.2947056.$$

In *Al-Mubaid et al* (Al-Mubaid *et al.*, 2006), the authors implicate more aspects in estimating the semantic similarity between two concepts in UMLS. In addition to the cross-ontological path length, this measure introduces a new aspect that is related to the common specificity. The intuition behind the common specificity is that pairs of concepts that lie at a lower level in the hierarchy share more information and so tend to be more similar than others lying at a higher level in the ontology. Using only the shortest path to assess similarity, resulting scores might be biased as measures ignore the positions of concepts in the hierarchy. Common specificity takes into account the level where treated concepts reside according to the following formula:

$$C_{spec}(c_1, c_2) = Dc - depth(msca(c_1, c_2)) \quad (39)$$

Where:

Dc is the depth of the cluster or the branch of the ontology where the most specific common abstraction (msca) of the concepts (c_1, c_2) resides. This is used to scale the depth of msca.

For example (see Figure 24), the concept 'Bacterial Infection' is the Msca of 'strep-throat' and 'tetanus'.

$$Sim_{nam}(c_1, c_2) = \log \left((Sim_{Rada} - 1)^\alpha * (C_{spec})^\beta + K \right) \quad (40)$$

Where:

$\alpha, \beta > 0$ are parameters that can be set to 1 allowing both features to contribute equally to the final similarity. Given the same example (using $k=1$): $Sim('oral\ thrush', 'athlete's\ foot') = \log((3 - 1) * (3 - 1) + 1) = 0.699$.

In **Mao et al** (Mao et al., 2002), the authors assume that the similarity between two concepts in an ontology is related to the distance between them as well as their positions in the hierarchy; a concept is more similar to his father than his grandfather in the hierarchy. The generality of a concept, according to the authors, is related to the number of his descendants or hyponyms. This similarity measure can be calculated through the following formula:

$$Sim_{Mao}(c_1, c_2) = \frac{C}{Sim_{Rada}(c_1, c_2) * \log(1 + D(c_1) + D(c_2))} \quad (41)$$

Where:

$D(c)$ is the number of descendants of the concept c .

C is a constant.

For example using ($C=1$), $Sim('oral\ thrush', 'athlete's\ foot') = \frac{1}{3 * \log(1+2+3)} = 0.428$

In **Zhong et al** (Zhong et al., 2002), the author propose a milestone for each node in the hierarchy as the following:

$$milestone(c) = \frac{1/2}{k^{depth(c)}} \quad (42)$$

Where:

$Depth(c)$ is the depth of the node c in the hierarchy

k is a constant that is usually set to 2. This constant is a predefined factor that indicates the rate at which the value decreases along the hierarchy.

According to this work the distance between two concepts is estimated using the previous milestone as the following:

$$dc(c1, c2) = dc(c1, msca(c1, c2)) + dc(c2, msca(c1, c2)) \quad (43)$$

Where:

$msca(c1, c2)$ is the closest common parent of $c1, c2$

$dc(c, msca) = milestone(msca) - milestone(c)$

$dc('oral\ thrush', 'athlete's\ foot')$

$$\begin{aligned} &= milestone('fungal\ infection') - milestone('oral\ thrush') \\ &+ milestone('fungal\ infection') - milestone('athlete's\ foot') \\ &= 0.25 + 0.0625 + 0.25 + 0.125 = 0.6875 \end{aligned}$$

4.1.3 Discussion

Previous subsections presented some Ontology-based similarity measures which depend on the structure of the ontology in assessing semantic similarities between its concepts. The presented measures are synthesized in Table 16. The measures belonging to this family are the simplest as compared to other families which is the main advantage of the preceding works. Some of them depend only on the *shortest path* (Rada et al., 1989; Bulskov et al., 2002; Caviedes et al., 2004) while others deploy the position and the depth of concepts or their msca in the hierarchy as well

as the path in assessing similarities in order to take concepts' specificity into account (Wu *et al.*, 1994; Leacock *et al.*, 1998; Mao *et al.*, 2002; Zhong *et al.*, 2002; Y. Li *et al.*, 2003; Al-Mubaid *et al.*, 2006).

Some of these measures chose a *linear function* to assess similarities between concepts (Rada *et al.*, 1989; Wu *et al.*, 1994; Bulskov *et al.*, 2002; Caviedes *et al.*, 2004) while others used *nonlinear function* that are argued to be more adequate to assess similarities (Leacock *et al.*, 1998; Mao *et al.*, 2002; Zhong *et al.*, 2002; Y. Li *et al.*, 2003; Al-Mubaid *et al.*, 2006). In fact, nonlinear functions can calculate semantic similarity and generate values in a specific range using characteristics that have infinite values like path length.

In fact, the main advantage of Ontology-based measures is their efficiency as compared to measures of other families. This efficiency is related to their simplicity and dependency on the structure of the ontology requiring no external knowledge resources. Nevertheless, this family requires consistent, big, and fine grained ontologies covering the application domain. Only two of the presented measures were developed for the medical domain (Mao *et al.*, 2002; Al-Mubaid *et al.*, 2006) whereas the others were applied to general purpose semantic resources.

Reference	Basic principle	Application resource/domain	Advantages	Disadvantages
Rada <i>et al.</i> (1989)	-Count IS-A links of the shortest path	-Estimate the similarity on semantic networks	-Simplicity	-Requires consistent ontologies
Bulskov <i>et al.</i> (2002)	-Count IS-A links of the shortest path -Longest path	-Document ranking -Query evaluation	-Simplicity	-Requires consistent ontologies
Wu <i>et al.</i> (1994)	-Depth of concepts -Path to the MSCA	-Machine translation for verb similarities	-Simplicity -Position of concepts is considered	-Requires consistent ontologies -Affected by the absence of a unique root node
Leacock <i>et al.</i> (1998)	-Counts nodes for the shortest path -depth of the ontology	-WordNet for word sense identification	-Simplicity -Log for smoothing	-Requires consistent ontologies -Affected by the absence of a unique root node
Y. Li <i>et al.</i> (2003)	-Shortest path -depth of MSCA	-WordNet	-Nonlinear function -Position of concepts is considered	-Requires consistent ontologies -Affected by the absence of a unique root node -Requires parameter tuning
Al-Mubaid <i>et al.</i> (2006)	-Depth of the ontology - Specificity of the MSCA	-MeSH+SNOMED -IR	-Log for smoothing -Common specificity -Local granularity	-Requires consistent ontologies -Affected by the absence of a unique root node -Requires parameter tuning
Mao <i>et al.</i> (2002)	-Depth of concepts -Shortest path	-UMLS -IR in the medical domain	-Log smoothing -Position of concepts is considered	-Requires consistent ontologies -Affected by the absence of a unique root node -Requires parameter tuning
Zhong <i>et al.</i> (2002)	-MSCA -Granularity and distance from the root node	-Conceptual graph matching	-Nonlinear function -Position of concepts is considered	-Requires consistent ontologies -Affected by the absence of a unique root node -Requires parameter tuning

Table 16. Structure-based similarity measures

4.2 Information theoretic measures

This family is also called *corpus-based measures*, as most measures depend on the statistics of concepts' occurrences in a particular corpus in order to obtain their Information Content (IC). Resulting values depend on the corpus and its particularity; changing the corpus would effect the similarity measure itself. Thus, many methods propose to calculate concepts' IC depending on the structure of the ontology. Here again, authors depend completely on the hierarchy in assessing semantic similarities. For example (see Figure 24), the root node 'infection' is less informative than the leave node 'oral rush' which has a very specific sense.

4.2.1 Computing IC-based semantic similarity measures using corpus statistics

According to **Resnik** (Resnik, 1995), the information content (IC) of a concept in the ontology is given by formula (44). So we attribute to each concept a value that is related to its occurrences in the corpus in use.

$$IC_{res}(c) = -\log(p(c)) \quad (44)$$

Where c is the considered concept, $p(c)$ is the probability of this concept c occurring in a particular corpus. The probability function $P(c): C \rightarrow [0,1]$, is defined as follows:

$$p(c) = \frac{freq(c)}{N} = \frac{\sum_{n \in words(c)} count(n)}{N} \quad (45)$$

Where:

N is the total number of words seen in the corpus

$words(c)$ is the group of words subsumed by the concept c .

Obviously, the more probable the concept's occurrence in the corpus, the less informative it is, so the more general a concept, the lower its IC.

The measures of this family use the IC derived from the preceding formulas and consider that its value summarizes and quantifies the semantic content of the concept. Forthcoming measures are focused on how to quantify the shared semantics or the semantic similarity between two concepts using their ICs values. The basic idea is that the semantic similarity resides in the IC of the most specific concept that subsumes both concepts.

Resnik measure (Resnik, 1995) assesses the semantic similarity between two concepts using the IC of their msca (Most Specific Common Abstraction); the shared information between two concepts is the maximum IC of their msca.

$$Sim_{Res}(c_1, c_2) = \max_{c \in S(c_1, c_2)} IC_{res}(c) \quad (46)$$

Where:

$S(c_1, c_2)$ is the group of shared parents between the compared concepts.

$IC_{res}(c)$ is the information content of concept $c = msca(c_1, c_2)$ according to the formula (44). The resulting values vary between 0 and $\log(N)$ where N is the size of the corpus.

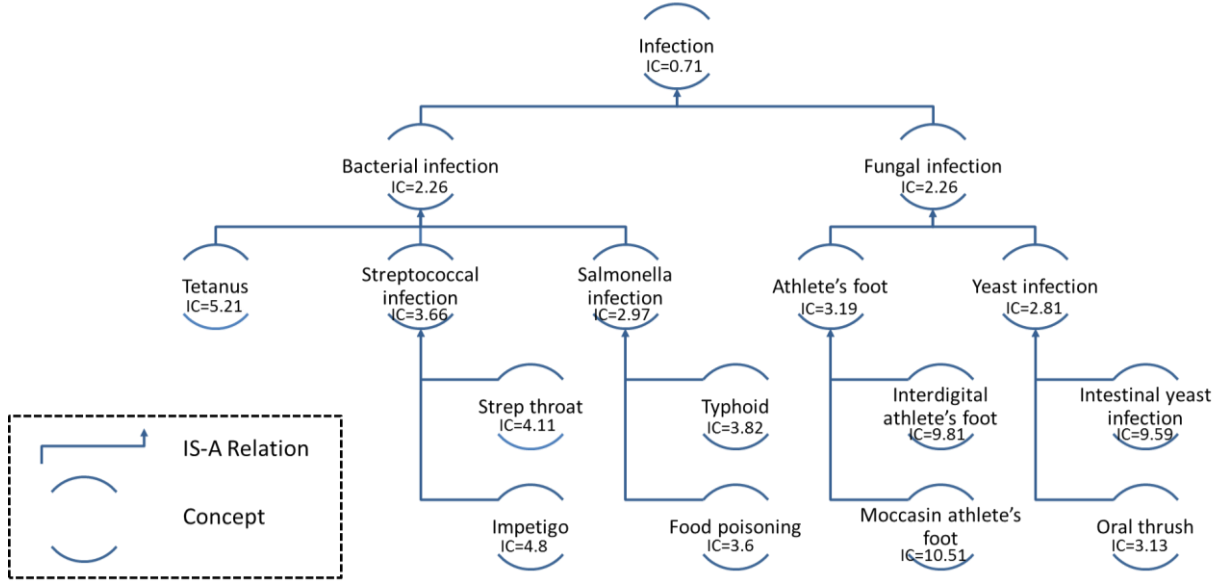


Figure 25. A part of UMLS IC of each concept is calculated using a medical corpus according to (Resnik, 1995; Pedersen et al., 2012)

Lin measure (Lin, 1998) uses the IC of both concepts in addition to the previous measure. This measure has a better ranking of similarities than the preceding one where different pairs of concepts can have the same $msca$ and consequently have the same similarity. Similarly to the structure-based measure according to **wu et al** (Wu et al., 1994), this measure is given in the following formula:

$$Sim_{Lin}(c_1, c_2) = \frac{2 * Sim_{Res}(c_1, c_2)}{IC(c_1) + IC(c_2)} = \frac{2 * IC(LCS)}{IC(c_1) + IC(c_2)} \quad (47)$$

Where:

$Sim_{Res}(c_1, c_2)$ is calculated according to the formula (46).

$IC(c_1)$ is calculated according to the formula (44).

Jiang measure (Jiang et al., 1997) is a semantic distance measure that is given in the following formula:

$$dist_{Jcn}(c_1, c_2) = (IC(c_1) + IC(c_2)) - 2 * Sim_{res}(c_1, c_2) \quad (48)$$

Where:

$Sim_{Res}(c_1, c_2)$ is calculated according to the formula (46)

$IC(c_1)$ is calculated according to the formula (44).

So the semantic similarity between two concepts is as follows:

$$Sim_{jiang}(c_1, c_2) = 1 - dist_{jcn}(c_1, c_2) \quad (49)$$

Similarly to the measure **Lin**, this measure combines the *IC* of the msca and the *ICs* of compared concepts. Resulting values are comparable to those in **Resnik** as it varies from 0 to $2 * \log(N)$. In fact, this measure combines the characteristics of both **Lin** and **Resnik** measures.

4.2.2 Computing IC-based semantic similarity measures using the ontology

According to previous methods, a collection of documents or a corpus can be used to calculate IC values for each concept in the ontology. Resulting values must respect the following condition:

for any pair of concepts (c_1, c_2) , if c_1 "IS - A" $c_2 \rightarrow IC(c_1) \geq IC(c_2)$.

However, even with large text corpora, the preceding condition is not always respected.

Ontology-based methods for IC, also called intrinsic methods, are based on the hypothesis that the ontology is an explicit model of the knowledge so the IC of its concepts can be directly derived from its hierarchy. Authors in (Pirro, 2009) argue that the ontology is structured and organized according to the principle of “*Cognitive Saliency*” that states that new concepts are created when the difference between them and the existing concepts is substantial.

According to the authors in (Seco *et al.*, 2004), we can avoid using a particular corpus to estimate the IC of concepts in a way to make it more generic and less expensive. In addition they argue that WordNet can also be used as a statistical resource with no need for external corpora. The intuition behind the hypothesis is that the more the hyponyms of a concept in the ontology the less informative it is considered. So the leaves in the hierarchy have the maximum IC ($IC_{c \text{leaves}}(c)=1$) which decreases as the concepts start to have more descendants or hyponyms in higher levels. Thereby, IC is a function of the population of a concept’s hyponyms as in the following formula:

$$IC_{Seco}(c) = \frac{\log\left(\frac{hypo(c) + 1}{max}\right)}{\log\left(\frac{1}{max}\right)} = 1 - \frac{\log(hypo(c) + 1)}{\log(max)} \quad (50)$$

Where:

hypo(c) is the number of hyponyms subsumed by the concept *c*

max is a constant that is usually set to the number of concepts in the ontology.

$\log(1/max)$ is used to ensure that resulting values range between [0,1].

For example (Figure 25):

$$IC_{Seco}(Salmonella Infection) = 1 - \frac{\log(2 + 1)}{\log(16)} = 0.61$$

The preceding formula guarantees that the IC of ontology’s concepts would decrease as we treat higher levels in the hierarchy until we arrive at the root where IC=0. In fact, this IC measure was applied to the previous similarity measures: Resnik, Lin, Jiang using respective formulas: (46), (47) and (49) and replacing IC(c) by the one proposed in (50). Test results showed better correlation with human judgments than the original ones.

According to authors in (Z. Zhou *et al.*, 2008), the position of the concept is also considered as a factor in calculating the IC it contains. So as in formula (51), the function $depth(c)$ represents the contribution of the position of the concept by incorporating its depth in the hierarchy. This is the major advantage of this approach as compared to the previous one.

$$IC_{Zhou}(c) = k \left(1 - \frac{\log(hypo(c) + 1)}{\log(max)} \right) + (1 - k) \left(\frac{\log(depth(c))}{\log(depth_{max})} \right) \quad (51)$$

Where:

$depth_{max}$ is the maximum depth of the ontology.

k is used to vary the contribution of each of the factors to the resulting information content.

For example (Figure 25):

$$IC_{Zhou}(Salmonella Infection) = \frac{1}{2(0.61)} + 1/2 \frac{\log(3)}{\log(4)} = 0.705$$

According to authors in (Sanchez, Batet, *et al.*, 2011), the number of leaves that the treated concept subsumes as well as the number of its ancestors are important factors in estimating its IC as the following formula:

$$IC_{Sanchez}(c) = -\log \left(\frac{\frac{|leaves(c)|}{|subsumer(c)|} + 1}{max_leaves + 1} \right) \quad (52)$$

Where:

$|leaves(c)|$ is the number of leaves subsumed by c

$|subsumer(c)|$ is the number of concepts that subsume the concept c

max_leaves is the number of leaves subsumed by the root node of the ontology.

For example (Figure 25):

$$IC_{Sanchez}(Salmonella Infection) = 1 - \log \frac{(2/2 + 1)}{9} = 0.65$$

In fact, this measure considers that concepts with many leaves in their hyponym tree are general (i.e., they have low IC) as they subsume the meaning of many terms. In addition, considering the number of subsumers of a concept introduces a broader and more realistic notion of concept's concreteness than previous measures that are based solely on the taxonomical depth; a concept that inherits from several subsumers is more specific than another one inheriting from a unique subsumer, even belonging to the same level of depth. Having several subsumers provides the concept with more distinctive features in order to differentiate it from its subsumers.

4.2.3 Discussion

This family of measures is based on the information theory in assessing semantic similarities between concepts. We identified two sources of information content (IC) for these measures: corpus (Resnik, 1995; Jiang *et al.*, 1997; Lin, 1998) and ontology structure (Seco *et al.*, 2004; Z. Zhou *et al.*, 2008; Sanchez, Batet, *et al.*, 2011). Thus, some semantic similarity measures depend on a corpus to implement IC theory while others, so called intrinsic measures, use the

structure of the ontology to assess the IC of its concepts. In fact, intrinsic measures can be considered as *hybrid approaches* combining ontology-based and IC-base principles. Table 17 synthesizes and compares the presented measures.

Measures based on corpus statistics are highly complex and require a corpus for collecting concepts' occurrences. Their dependency on the corpus in assessing the IC of the concepts is their main drawback. In fact, corpora sparseness and size affect these measures especially their accuracy and processing time. Moreover, corpus-based IC measures do not guarantee that a concept's IC is inferior to those of its children. These drawbacks were overcome in intrinsic or ontology-based approaches that consider ontologies as complete and explicit knowledge models. Nevertheless, intrinsic measures require consistent, fine-grained and well-structured Ontologies that provide a complete explicit representation of the application domain.

All of the presented IC-based semantic similarity approaches were applied tested on WordNet which is a general purpose ontology and some of them are adapted to the medical domain and implemented to assess the semantic similarity between concepts in UMLS (Pedersen *et al.*, 2012).

Type	Reference	Basic principle	Application resource/domain	Advantages	Disadvantages
Corpus statistics	Resnik (1995)	-IC of the MSCA	-IS-A Taxonomy -WordNet	-Simplicity	-Depends on the corpus -Doesn't respect that parent's IC is less than those of his children -Any pair that share the same msca have the same similarity
	Lin (1998)	-IC of the MSCA -IC of the compared concepts	-IS-A Taxonomy -WordNet	-Takes into consideration the IC of the compared concepts	-Depends on the corpus -Doesn't respect that parent's IC is less than those of his children
	Jiang <i>et al.</i> (1997)	-IC of the MSCA -IC of the compared concepts	-IS-A Taxonomy -WordNet	-Takes into consideration the IC of the compared concepts	-Depends on the corpus -Doesn't respect that parent's IC is less than those of his children
Ontology structure	Seco <i>et al.</i> (2004)	-IC is related to the number of descendants	-WordNet	-Respects that parent's IC is less than those of his children -Better correlation with human judgment: applied to Resnik, Lin, Jiang	-Depends on the ontology -Requires consistent, fine grained and well-structured Ontologies that are representative of the domain -Discards the position of concepts in the hierarchy in IC formula
	Z. Zhou <i>et al.</i> (2008)	-IC is related to the number of descendants -IC is related to the depth of the node as well	-WordNet	-Respects that parent's IC is less than those of his children -Takes into consideration the position of the concept in the hierarchy -Better correlation with human judgment: applied to Resnik, Lin, Jiang	-Depends on the ontology -Requires consistent, fine grained and well-structured Ontologies that are representative of the domain -Requires parameter tuning
	Sanchez, Batet, <i>et al.</i> (2011)	-IC is related to the number of subsumers -IC is related to the leaves the node subsumes	-WordNet	-Respects that parent's IC is less than those of his children -Introduces concreteness more realistic than depth as it takes into consideration multiple inheritances -Better correlation with human judgment: applied to Resnik, Lin, Jiang	-Depends on the ontology -Requires consistent, fine grained and well-structured Ontologies that are representative of the domain

Table 17. IC-based similarity measures

4.3 Feature-based measures

These measures use the characteristics of the compared concepts in order to assess the similarity among them ignoring their positions in the ontology as well as their information content. While the structure-based measures seem to be unable to assess the semantic similarity between concepts of separated ontologies, feature-based measures represent a good solution. In fact, this category is considered the most general as compared to the preceding two categories (Petrakis *et al.*, 2006). Next subsection introduces the vision of Tversky which represents the main inspiration of the measures of the second subsection.

4.3.1 The vision of Tversky

The measures of this category assess the semantic similarity between two concepts using their descriptive properties assuming that each concept is described by a group of words that are related to its characteristics. The more the characteristics the compared concepts have in common the more similar they are considered and vice versa (Tversky, 1977).

Defining these characteristics is very critical issue to these measures. The existing approaches try to define these characteristics in different manners; some of them use the information delivered in the ontology in terms of synonyms (as the synsets in WordNet); some of them use the textual definitions of the concepts in the ontology (called glosses in WordNet); others use also the different types of semantic relations in the ontology (Sanchez *et al.*, 2012). As these characteristics can be found in the context of the concept, this category is also called *context-based measures*.

Given the concepts “Car” and “Bicycle”, as it is illustrated in Figure 26, both concepts are hyponyms of the concept “Wheeled vehicle” and they are so related to it by an IS-A relation. Thus both concepts share the characteristics that are related to “Wheeled vehicle” in general, such as having the concepts “wheel” and “brake” related to them by “Part-Of” relation. On the other hand, each of these two concepts has its particular characteristics that discriminate it from other wheeled vehicles. In a conclusion, a *feature based similarity measure* might take into consideration all of the relations connecting both concepts to others in the ontology in order to estimate the similarity among them.

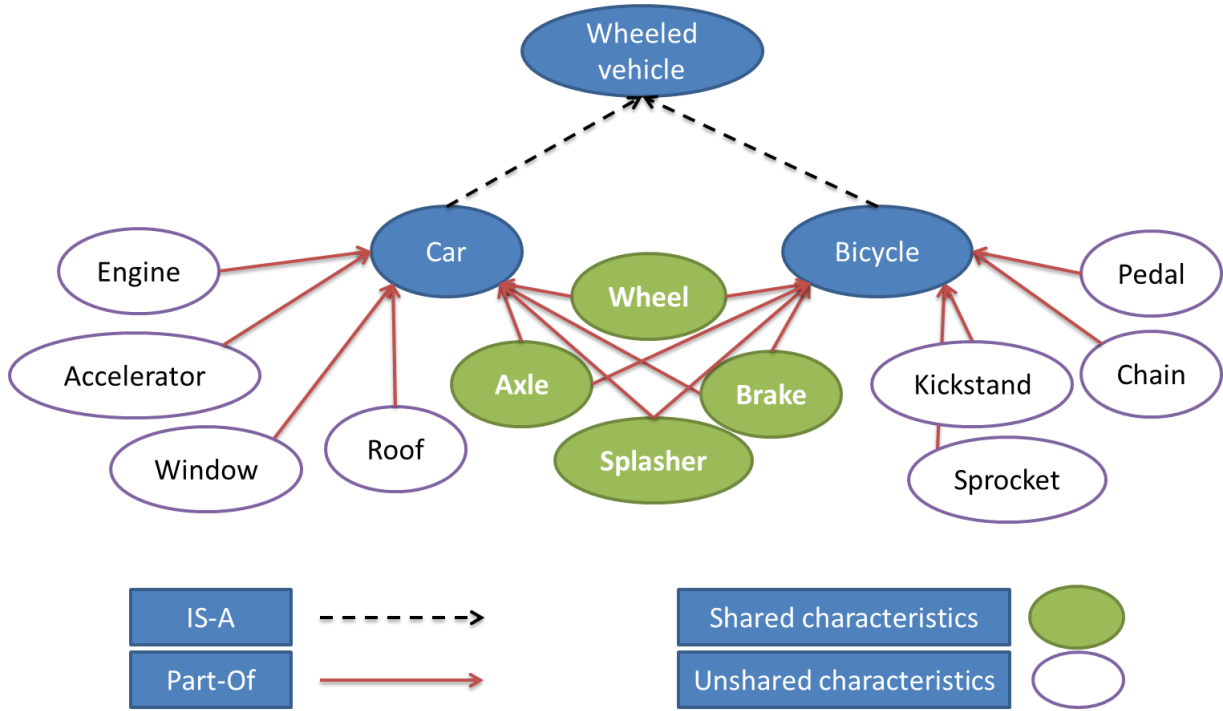


Figure 26. Common characteristics among two concepts

4.3.2 Feature-based semantic similarity measures

This family of measures applies the vision of Tversky in the context of semantic similarity in order to assess the semantic common features among a pair of concepts in a semantic knowledge base.

Tversky (Tversky, 1977) proposed a general model to estimate the similarity among two concepts c_1 and c_2 . Using the sets of descriptive words of these concepts as $\psi(c_1)$ and $\psi(c_2)$, Tversky defines the shared characteristics among them as the intersection of these sets ($\psi(c_1) \cap \psi(c_2)$). Furthermore, the resulting set of $(\psi(c_1) \setminus \psi(c_2))$ represents the characteristics of c_1 that are not shared with c_2 . These sets are illustrated in Figure 27.

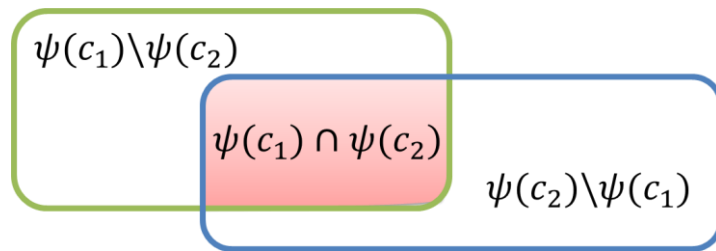


Figure 27. Sets of common and distinctive characteristics of concepts C_1 , C_2 .

According to the preceding definitions, Tversky proposed the following model to assess the similarity between two concepts c_1 and c_2 :

$$Sim_{Tvr}(c_1, c_2) = \alpha F(\psi(c_1) \cap \psi(c_2)) - \beta F(\psi(c_1) \setminus \psi(c_2)) - \gamma F(\psi(c_2) \setminus \psi(c_1)) \quad (53)$$

Where:

F is a general function that considers the number of characteristics in a particular set.

α , β and γ are parameters that control the contribution of each of the three factors to the formula.

These parameters define the role of common and distinctive characteristics in similarity judgment. In order to restrict the values of the similarity measure to this range $[0,1]$ no matters the size of characteristic sets, this measure assesses the similarity according to the following formula:

$$Sim_{Tvr-ratio}(c_1, c_2) = \frac{\alpha F(\psi(c_1) \cap \psi(c_2))}{\beta F(\psi(c_1) \setminus \psi(c_2)) + \gamma F(\psi(c_2) \setminus \psi(c_1)) + \alpha F(\psi(c_1) \cap \psi(c_2))} \quad (54)$$

Where:

$\alpha=1$ giving the maximum contribution to the common characteristics.

For a symmetric measure where: $sim(c_1, c_2) = sim(c_2, c_1)$, the values of β and γ must be equally tuned otherwise the equality condition of symmetry isn't respected. Table 18 illustrates the different scenarios where Tversky semantic similarity can be applied according to the tuning of parameters β and γ .

Case	Parameters	Description
Only common characteristics between c_1, c_2	$B=\gamma=0$	In case of any commonality: $Sim_{Tvr-ratio}(c_1, c_2) = 1$
Given c_1 assess to what extent c_2 is similar to it	$B=1, \gamma=0$	If $\psi(c_1) \subseteq \psi(c_2)$ then $Sim_{Tvr-ratio}(c_1, c_2) = \frac{\alpha F(\psi(c_1) \cap \psi(c_2))}{\gamma F(\psi(c_2) \setminus \psi(c_1)) + \alpha F(\psi(c_1) \cap \psi(c_2))}$
Given c_2 assess to what extent c_1 is similar to it	$B=0, \gamma=1$	If $\psi(c_1) \supseteq \psi(c_2)$ then $Sim_{Tvr-ratio}(c_1, c_2) = \frac{\alpha F(\psi(c_1) \cap \psi(c_2))}{\beta F(\psi(c_1) \setminus \psi(c_2)) + \alpha F(\psi(c_1) \cap \psi(c_2))}$
Given c_1, c_2 assess the similarity among them	$B=\gamma=1$	Tanimoto Index
	$B=\gamma=0,5$	Dice Index

Table 18. Different scenarios of Tversky similarity measure

Petrakis (Petrakis *et al.*, 2006) proposed a specific feature-based measure for WordNet depending on its structure particularity. In fact, WordNet is composed of a set of synsets each of which contains a set of synonyms. Thus, the proposed measure *X-Similarity* considers a set of synonyms or the term description set as the set of characteristics of the related concept. A term description set can be extracted from term definition in the ontology ("gloss" in WordNet). Authors also define the similarity between two characteristic sets as the following:

$$S_{synsets}(c1, c2) = \frac{|A \cap B|}{|A \cup B|} \quad (55)$$

Where:

A, B are sets of synonyms of $c1, c2$ respectively.

Authors (Petrakis *et al.*, 2006) propose a similar matching scheme for term description sets and the following for matching synsets of the neighbors of $c1, c2$:

$$S_{Neighbors}(c1, c2) = \max \frac{|A_i \cap B_i|}{|A_i \cup B_i|} \quad (56)$$

Where:

A_i, B_i are sets of synonyms of the neighbor i of c_1, c_2 respectively.

The preceding matching schemes are all combined in the X -Similarity measure as the following:

$$Sim_{X-sim}(c_1, c_2) = \begin{cases} 1, & \text{if } S_{synsets}(c_1, c_2) > 0 \\ \max\{S_{neighbors}(c_1, c_2), S_{descriptions}(c_1, c_2)\}, & \text{if } S_{synsets}(c_1, c_2) = 0 \end{cases} \quad (57)$$

Thus, two concepts are considered similar if their synsets or descriptions sets or the synsets of their neighbors are similar. This measure was applied on both WordNet and MeSH ontology for assessing cross ontology similarities and proved high correlation with human judgment (Petrakis *et al.*, 2006). For example, given the concept “Hypothyroidism” and the concept “Hyperthyroidism” from the ontologies WordNet and MeSH respectively, $S_{synsets} = 0$ since they have no synsets in common while $\max\{S_{neighbors}, S_{descriptions}\} = 0.387$.

WordNet term: Hypothyroidism	MeSH term: Hyperthyroidism
<code><term> hypothyroidism</code> <code><definition></code> An underactive thyroid gland; a glandular disorder Resulting from insufficient production of thyroid hormones. <code></definition></code> <code><synset></code> Hypothyroidism <code></synset></code> <code><hypernyms></code> glandular disease, disorder, condition, state <code></hypernyms></code> <code><hyponyms></code> myxedema, cretinism <code></hyponyms></code> <code></term></code>	<code><term> hyperthyroidism</code> <code><definition></code> Hypersecretion of Thyroid Hormones from Thyroid Gland. Elevated levels of thyroid hormones increase Basal Metabolic Rate. <code></definition></code> <code><synset></code> Hyperthyroidism <code></synset></code> <code><hypernyms></code> disease, thyroid, Endocrine System Diseases, diseases <code></hypernyms></code> <code><hyponyms></code> thyrotoxicosis, thyrotoxicoses <code></hyponyms></code> <code></term></code>

Table 19. XML descriptions of “Hypothyroidism” and “Hyperthyroidism” from WordNet and MeSH (Petrakis *et al.*, 2006)

Banerjee (Banerjee *et al.*, 2003) proposed a measure for assessing the semantic relatedness between two concepts based on the overlap or the shared words between their definitions or glosses respectively. According to this measure, Concepts do not need to be connected via relations or paths to measure the relatedness among their glosses which distinguish relatedness measures from other similarity measures. The measure proposed in (Banerjee *et al.*, 2003) extends the one proposed earlier in (Lesk, 1986) that is based on the hypothesis that “the more overlaps between two senses, the more related”. The extended approach involves the hypernyms and the hyponyms as well in assessing the semantic relatedness according to the following formula:

$$\begin{aligned}
Relatedness_{Banerjee}(c_1, c_2) = & \quad (58) \\
& score(Gloss(c_1), Gloss(c_2)) + score(hypernym(c_1), hypernym(c_2)) + \\
& score(hyponym(c_1), hyponym(c_2)) + score(Gloss(c_1), hypernym(c_2)) + \\
& score(hypernym(c_1), Gloss(c_2))
\end{aligned}$$

Note that the score is an accumulation of the squared sizes of all found overlaps between two compared glosses. The final score determines the relatedness between two concepts so the more overlaps between two concepts, the more related. For example (Banerjee *et al.*, 2003) *drawing paper* and *decal* have the glosses “*paper that is specially prepared for use in drafting*” and “*the art of transferring designs from specially prepared paper to a wood or glass or metal surface*”. We observe two overlaps: the single word “*paper*” and the two-word phrase “*specially prepared*” which results in: $1^2 + 2^2 = 5$ as the overlap score.

The same measure was extended in (Patwardhan *et al.*, 2006) using co-occurrences vectors of words in the gloss extracted by means of an external corpus. For each concept, a list of co-occurrences vectors is constituted depending on a plain text external corpus. The vectors of the list of each concept are averaged resulting in two vectors representing new definitions of the compared concepts. Finally, relatedness score is the cosine of the angle between these vectors as described earlier in chapter 2. The main advantage of this measure as compared with the previous one is that retrieving co-occurrences behavior of glosses’ words completes the glosses. In fact, it is difficult to measure relatedness depending on glosses only because of their brevity and the use of different synonyms in similar definitions. Authors in (Patwardhan *et al.*, 2006) believe that synonyms that are used in different glosses will tend to have similar Vectors as they usually show similar co-occurrence behavior.

4.3.3 Discussion

Feature-based measures assess the semantic similarity between two concepts by applying the vision of *Tversky* in different manners. The basic hypothesis is to assess the similarity by matching the feature sets of compared concepts. The details of the presented measures are synthesized in Table 20.

Most of the presented measures use glosses or synsets (of WordNet) of the ontology to constitute the feature sets of its concepts. In fact, these measures show strong dependency on the used ontology and the integrity of its glosses. Nevertheless, these measures are capable of cross ontology comparison which is not possible with structure or information-based similarity measures.

According to the literature, some measures belonging to this family used principles from preceding families in order to define the common and the distinctive features of the compared concepts. This might imply dependencies on sources of information other than the used ontology. We will discuss some of them in the next section that is focused on hybrid similarity measures.

Reference	Basic principle	Application resource/domain	Advantages	Disadvantages
Tversky (1977)	-Descriptive feature set matching -Common/distinctive features	-Comparing objects	-General model -Useful in IR and Clustering	-Objects should have descriptive sets
Petrakis et al. (2006)	Commonality between: -Synsets of terms -Their descriptive term sets -Synsets of their neighbors	-WordNet -MeSH	-Application of Tversky on WordNet -Requires no external knowledge -Uses Glosses -Cross ontology comparisons	-Depends on WordNet and MeSH
Banerjee et al. (2003)	-Overlaps between glosses of concepts, their hyponyms and their hypernyms	-WordNet -UMLS -WSD	-Requires no external knowledge -Uses Glosses -Cross ontology comparisons -Adaptability to different ontologies -Compare words in different POS	-Requires ontologies with complete glosses
Patwardhan et al. (2006)	-Glosses -Co-occurrences vectors of glosses' words from a corpus -Cosine similarity	-WordNet -UMLS	-Uses Glosses -Cross ontology comparisons -Adaptability to different ontologies -Compare words in different POS -Complete Glosses with co-occurrences observed in a corpus	-Requires ontologies with complete glosses -Requires plain text corpus for co-occurrences

Table 20. Feature-based similarity measures

4.4 Hybrid measures

This family of measures combines the principles of two or more measures from different preceding families. In fact, these measures tend to combine the advantages of these families and to avoid their weak points as well.

4.4.1 Some hybrid measures

Knappe (Knappe *et al.*, 2007) proposed a structure and feature-based similarity measure. The main aspect treated in this measure is that there may be multiple paths connecting two concepts. Taking all possible paths into consideration increases the complexity substantially. Instead of traversing the possible paths, shared concepts are a good solution. Authors also introduce the notion of *term decomposition* where the compared term is decomposed into a set of concepts. For each concept of the set, upward expansion determines the related generalizations of the concept. Finally, for each initial term, a graph of related concept is derived from the ontology and so the similarity between them is given in the following formula:

$$Sim_{Knappe}(c_1, c_2) = \rho \frac{|u(c_1) \cap u(c_2)|}{|u(c_1)|} + (1 - \rho) \frac{|u(c_1) \cap u(c_2)|}{|u(c_2)|} \quad (59)$$

Where:

$\rho \in [0,1]$ denotes the degree of influence of generalizations

$|u(c_1) \cap u(c_2)|$ denotes the number of reachable nodes shared by c_1 and c_2

Pirro (Pirro *et al.*, 2010) proposed an IC-based application of the *Tversky* (Tversky, 1977) feature based model for similarity. The main assumption is that the IC decreases monotonically as we move from the leaves to the root of a taxonomy. Starting from this assumption, we can infer the common and the distinctive features of concepts. Given the concept “car” from Figure 26, we can estimate the set of distinctive features of “car” is $IC_{car} - IC_{wheeled\ vehicle}$ and respectively for “bicycle” $IC_{bicycle} - IC_{wheeled\ vehicle}$. As for common features they can be replaced by the $IC_{wheeled\ vehicle}$ which is the IC of the msca of “car” and “bicycle”. These mappings are generalized in Table 21.

Description	Feature-based model	Information-theoretic model
Common features	$\psi(c_1) \cap \psi(c_2)$	$IC(msca(c_1, c_2))$
Features of C_1 alone	$\psi(c_1) \setminus \psi(c_2)$	$IC(c_1) - IC(msca(c_1, c_2))$
Features of C_2 alone	$\psi(c_2) \setminus \psi(c_1)$	$IC(c_2) - IC(msca(c_1, c_2))$

Table 21. Mapping between feature-based and IC similarity models (Pirro *et al.*, 2010)

Authors in (Pirro *et al.*, 2010) proposed *Feature and Information Theoretic* (FaITH) measure for semantic similarity and relatedness adopting previous mappings in the model of *Tversky* as the following:

$$Sim_{Faith} = \frac{IC(msca(c_1, c_2))}{IC(c_1) + IC(c_2) - IC(msca(c_1, c_2))} \quad (60)$$

An extended IC measure is also proposed combining two different values. The first one is the average of IC values for all concepts $c_k \in C_{R_j}$ that are related to the treated concept with m types of relations:

$$EIC(c) = \sum_{j=1}^m \frac{\sum_{k=1}^n IC(c_k \in C_{R_j})}{|C_{R_j}|} \quad (61)$$

Where:

$|C_{R_j}|$ is the number of related concepts.

The second value iIC is the information content given by formula (50). Thus, the *extended Information Content* measure (eIC) is calculated as the following:

$$eIC(c) = \zeta iIC(c) + \eta EIC(c) \quad (62)$$

Where:

ζ, η are used to weight the contribution of the iIC and EIC coefficients.

This measure was tested on a collection of pairs of concepts retrieved from WordNet and MeSH and showed better correlation with human judgment than classical structure-based and IC-based similarity measures (Pirro *et al.*, 2010).

Authors in (Sanchez, & Batet, 2011) proposed a mapping from well-known similarity coefficients in the set theory into IC based similarity measures (Table 22). These mappings are based on the mappings formerly presented in Table 21. These measures showed good correlation with human judgment when compared with classical measures (Sanchez, & Batet, 2011).

Function	Original formula	IC-based formula
Jaccard (63)	$\frac{ c_1 \cap c_2 }{ c_1 \cup c_2 }$	$\frac{IC(msca(c_1, c_2))}{IC(c_1) + IC(c_2) - IC(msca(c_1, c_2))}$
Dice (64)	$\frac{2 * c_1 \cap c_2 }{ c_1 + c_2 }$	$\frac{2 * IC(msca(c_1, c_2))}{IC(c_1) + IC(c_2)}$
Ochiai (65)	$\frac{ c_1 \cap c_2 }{\sqrt{ c_1 * c_2 }}$	$\frac{IC(msca(c_1, c_2))}{\sqrt{IC(c_1) * IC(c_2)}}$
Simpson (66)	$\frac{ c_1 \cap c_2 }{Min(c_1 , c_2)}$	$\frac{IC(msca(c_1, c_2))}{Min(IC(c_1), IC(c_2))}$
Braun-Blanquet (67)	$\frac{ c_1 \cap c_2 }{Max(c_1 , c_2)}$	$\frac{IC(msca(c_1, c_2))}{Max(IC(c_1), IC(c_2))}$
Sokal et Sneath (68)	$\frac{ c_1 \cap c_2 }{2 * (c_1 + c_2) - 3 * c_1 \cap c_2 }$	$\frac{IC(msca(c_1, c_2))}{2 * (IC(c_1) + IC(c_2)) - 3 * IC(msca(c_1, c_2))}$

Table 22. Mapping between set-based similarity coefficients and IC-based coefficients

Sanchez (Sanchez *et al.*, 2012) proposed a new dissimilarity measure as an application of *Tversky* vision. We identified two major differences between this measure and others applying the same vision: First, authors propose a feature-based measure that doesn't require parameter tuning or a corpus for feature extraction. In fact, authors consider subsumers as the set features of the treated concept; subsumers are labels that describe the meaning of the concept in different levels of generality. Second, logarithm smoothing of the ratio is a nonlinear function that is argued to be more adequate to evaluate features (Leacock *et al.*, 1998; Y. Li *et al.*, 2003; Al-Mubaid *et al.*, 2006) and also to approximate better the concept of similarity.

$$DisSim_{Sanchez}(c_1, c_2) = \log_2 \left(1 + \frac{|\emptyset(c_1) \setminus \emptyset(c_2)| + |\emptyset(c_2) \setminus \emptyset(c_1)|}{|\emptyset(c_1) \setminus \emptyset(c_2)| + |\emptyset(c_2) \setminus \emptyset(c_1)| + |\emptyset(c_1) \cap \emptyset(c_2)|} \right) \quad (69)$$

Where:

$|\emptyset(c_1) \setminus \emptyset(c_2)| + |\emptyset(c_2) \setminus \emptyset(c_1)|$ is the number of uncommon features of compared concepts

$|\emptyset(c_1) \setminus \emptyset(c_2)| + |\emptyset(c_2) \setminus \emptyset(c_1)| + |\emptyset(c_1) \cap \emptyset(c_2)|$ is the total number of features of both concepts which is used to scale the previous value.

Given two concepts "Sailing", "Sunbathing" with the following sets of characteristics:

$\emptyset(\text{Sailing}) = \{\text{sailing, wind, water, sport, activity, beach}_{leisure}\}$

$\emptyset(\text{Sunbathing}) = \{\text{sunbathing, activity, beach}_{leisure}\}$

According to the proposed measure in (Sanchez *et al.*, 2012) the dissimilarity between these concepts is assessed as follows:

$$DisSim_{Sanchez}(\text{"Sailing"}, \text{"Sunbathing"}) = \log_2 \left(1 + \frac{4 + 1}{4 + 1 + 2} \right) = 0.78$$

4.4.2 Discussion

This section presented some hybrid semantic similarity measures that combine the principles of two or three of the preceding families. The presented measures are synthesized in Table 23.

Most measures combined structure-based or IC-based principles with feature-based principles constituting hybrid approaches. Obviously, combining different principles increases the complexity of the resulting measure which is the major drawback of this family. However, these measures combine the advantages of the underlying principles and overcome their limitation. In fact, experimental studies confirmed this assumption and demonstrated that the measures of this family have better correlation with human judgment than other families.

Structure	Type		Reference	Basic principle	Application resource/domain	Advantages	Disadvantages
	IC	Feature					
X		X	Knappe <i>et al.</i> (2007)	-Commonalities among sets of upward reachable nodes for compared concepts	Ontology-based querying	-Requires no external knowledge -Takes all reachable paths into consideration -Uses the knowledge in the structure of the ontology to assess similarity	-Average complexity -Requires consistent ontologies -Requires parameter tuning
	X	X	Pirro <i>et al.</i> (2010)	-IC of compared concepts and their msca -The average IC of the related concept of each type of relations	-WordNet -MeSH	-Structure based IC -Takes into consideration relations other than IS-A	-Average complexity -Requires consistent ontologies -Requires parameter tuning
	X	X	Sanchez, and Batet (2011)	-IC of compared concepts and their msca -Classical set based similarity measures	-Biomedical domain (SNOMED)	-Structure based IC -Inspired from Set theory coefficients of similarity	-Average complexity -Requires consistent ontologies
X		X	Sanchez <i>et al.</i> (2012)	-subsumers of a concept constitute its feature set	WordNet	-Logarithm smoothing -Requires no external knowledge sources -Uses subsumers as features on different levels of generalization	-Average complexity -Requires consistent ontologies

Table 23. Hybrid similarity measures

4.5 Comparing families of semantic similarity measures

In previous subsections, we presented three main families of semantic similarity measures (Ontology-based measures, IC-based measures and Feature-based measures). In addition, we presented some hybrid measures combining principles from different families in order to combine their advantages and to limit their disadvantages. Table 24 synthesizes the major characteristics of these three families.

The most attractive advantage of ontology-based measures is their simplicity. This applies also on intrinsic IC-based measures and Feature-based measures that extract features from the ontology which results in moderate complexity. In fact, depending only on the structure of the ontology minimizes the cost of similarity calculation (Sanchez *et al.*, 2012). Nevertheless, a number of limitations of these measures are well identified in the literature. First of all, for most of these measures only the shortest path between the treated concepts counts. Second, they consider that all IS-A links in taxonomy are defined by the same distance which requires consistent and fine grained ontologies (Pirro *et al.*, 2010).

Family	Basic assumption	Dependencies	Calculation unit	Advantages	Disadvantages
Ontology-based	Spreading activation theory	Ontology	-Shortest path -Depth/Granularity -msca	-Simplicity -efficiency -Requires no external knowledge sources	-Requires consistent ontologies
IC-based	Information content theory	Corpus	-Logarithm of occurrences probability -msca	-Uses linguistic statistics rather than positions in ontologies	-High complexity -Requires representative corpus
		Ontology	-N° of Hypernyms -N° of Hyponyms -N° of Leaves -Depth	-Requires no external knowledge sources other than ontologies	-Requires consistent ontologies -Moderate complexity
Feature-based	Tversky vision	Ontology/Corpus	Commonalities Between: -Synsets/Glosses -Hypernyms	-Cross ontology similarity measures -Adaptability to different ontologies	-Requires complete glosses/synsets -Parameter tuning

Table 24. Comparison between Structure, IC, and Feature-based similarity measures

In general, measures that exploit additional semantic evidence demonstrated higher accuracies like corpus base IC measures. In fact, IC-based measures capture implicit semantics in plain text as a function of frequency distribution in corpora. Nevertheless, the mapping between the words as observed in plain text and concepts is not straightforward and requires sense disambiguation. Moreover, these measures are affected by corpora availability and their sparseness (Seco *et al.*, 2004; Sanchez *et al.*, 2012).

Feature-based approaches, the only family that provides cross ontology comparisons, rely on features which are hardly found in domain ontologies, such as non-taxonomic relationships, attributes, synonym sets or glosses. Thus, their high dependency on information

availability affects its accuracy (Sanchez *et al.*, 2012). Finally, hybrid measures use ontology or IC-based principles to extract features overcoming previous limitations. Nevertheless, their high complexity increases the difficulty when adopting them in large scale applications.

5 Conclusion

This chapter presented first an introduction to semantics through its principle notions and some conventional semantic resources that were the center of interest of many researches in the domain of IR and text classification as well. Then, it presented works in the literature deploying semantics and semantic resources in text classification and other tasks related to IR intending to improve effectiveness. Different levels of semantic integration were investigated in section 3 starting from text representation to classification model and finally the class prediction or text to text comparison. Many of these approaches reported significant improvement in effectiveness after integrating semantics. Moreover, many authors underlined problems related to specific domains and particularly when dealing with the medical domain and argued the utility of using specific domain ontologies instead of general purpose ones in such contexts.

According to the literature, most works investigated the effect of semantics on text treatment at the representation level after indexing (Caropreso *et al.*, 2001; Liu *et al.*, 2004; Bloehdorn *et al.*, 2007; Séaghdha *et al.*, 2008; Wang *et al.*, 2008; Aseervatham *et al.*, 2009; Z. Li *et al.*, 2009; Séaghdha, 2009). In general, most works deployed explicit semantics as specified in ontologies through concepts generating *BOC* as a model for text representation (Bloehdorn *et al.*, 2006; Hliaoutakis *et al.*, 2006; Mihalcea *et al.*, 2006; Gabrilovich *et al.*, 2007; Guisse *et al.*, 2009; Bai *et al.*, 2010; L. Huang *et al.*, 2012). *Conceptualization* is the process of mapping text to concepts that we intent to deploy in order to enrich the original BOW and to overcome its three limitations: redundancy, ambiguity and orthogonality. Results of tests deploying explicit semantics in the literature are promising as they demonstrated improvements in classification.

We were also interested in this chapter in different state of the art similarity measures that assess the semantic similarity between pairs of concepts in an ontology. In fact, this similarity is the foundation for many approaches intending to use semantics in text representation and also in assessing text-to-text similarity. In fact, many state of the art works deployed semantic similarity between concepts in order to enriched text representation using *semantic kernels* with SVM classifiers (Bloehdorn *et al.*, 2007; Wang *et al.*, 2008; Aseervatham *et al.*, 2009; Séaghdha, 2009) while others used *generalization* in order to involve superconcepts in text representation (Bloehdorn *et al.*, 2006). Authors in (L. Huang *et al.*, 2012) proposed a method to enrich compared documents mutually using semantic similarities among their concepts. Considering text-to-text semantic similarity, few works proposed measures that involve semantics in class prediction. Most approaches are aggregation functions on semantic similarity between concepts pair-to-pair (Hliaoutakis *et al.*, 2006; Mihalcea *et al.*, 2006; Guisse *et al.*, 2009; L. Huang *et al.*, 2012). These approaches were developed in an ad hoc manner and need to be tested in large scale applications (L. Huang *et al.*, 2012).

Some authors went beyond the use of concepts and relations between them in the classification process; they used the entire hierarchy or parts of it as a representation model, a

classification model and a basis for prediction (Peng *et al.*, 2005; J. Z. Wang *et al.*, 2007; Guisse *et al.*, 2009). The intensive use of semantic resource structure can affect the efficiency of classification which makes enriching representation with similar concepts more advantageous.

All in all, reviewing state of the art works in this chapter aims to answer two questions: why integrating semantics in text classification is important and how to estimate its influence on classification effectiveness. Despite the promising results, the utility of integrating semantics in classification remains a subject of debate (Stein *et al.*, 2006). Nevertheless, it seems to be promising to take the application domain into consideration when developing a system for semantic classification (Ferretti *et al.*, 2008). In fact, further studies are warranted in order to determine the usefulness of semantics in text representation, training and class prediction. This will be the main focus of next chapters. We will propose generic testbeds to support semantic integration at different levels in next chapter intending to apply it to text classification in the medical domain.

CHAPTER 4: A FRAMEWORK FOR SUPERVISED SEMANTIC TEXT CLASSIFICATION

Table of contents

1	Introduction.....	111
2	Involving semantics in supervised text classification: a conceptual framework	112
3	Involving semantics through text conceptualization.....	114
3.1	Text Conceptualization Task	114
3.1.1	Text Conceptualization Strategies	114
3.1.2	Disambiguation Strategies.....	115
3.2	Generic framework for text conceptualization	116
3.3	Conclusion	116
4	Involving semantic similarity in supervised text classification	117
4.1	Semantic similarity.....	117
4.2	Proximity matrix.....	118
4.3	Semantic kernels.....	119
4.4	Enriching vectors.....	120
4.5	Semantic measures for text-to-text similarity	123
4.6	Conclusion	125
5	Methodology	127
5.1	Scenario 1: Conceptualization only	127
5.2	Scenario 2: Conceptualization and enrichment before training	127
5.3	Scenario 3: Conceptualization and enrichment before prediction	128
5.4	Scenario 4: Conceptualization and semantic text-to-text similarity for prediction	129
5.5	Conclusion	129
6	Related tools in the medical domain	131
6.1	Tools for text to concept mapping.....	131
6.1.1	PubMed Automatic Term Mapping (ATM).....	131
6.1.2	MaxMatcher	131
6.1.3	MGREP.....	132
6.1.4	MetaMap	132
6.2	Tools for semantic similarity	134
6.2.1	Semantic similarity engine	134
6.2.2	UMLS::Similarity.....	135
6.3	Conclusion	136
7	Conclusion	138

1 Introduction

Previous chapter presented a review on some state of the art works that studied the influence of semantics on supervised text classification and other tasks in the domain of information retrieval. Most authors gave experimental proofs that using semantics in indexing, in the classification model (and / or) in class prediction can improve classification effectiveness. In this chapter, we intend to present a generic framework for supervised semantic text classification involving semantics at different steps of text treatment. Next chapter implements this framework in an experimental platform in the intent to answer questions on the utility of semantics in the classification process.

The rest of this chapter is organized as the following: section 2 presents a conceptual framework for involving semantics in text classification at different steps of text classification. Section 3 presents specifications for involving semantics in text representation through *Conceptualization* and *Disambiguation*. Section 4 focuses on deploying semantic similarity measures in addition to concepts in text classification through *Representation Enrichment* and *Semantic Text-To-Text Similarity*, all using proximity matrix. Section 5 presents the *methodology* using which we intend to carry out the experimental study in next chapter. Here, we identify four different scenarios. Section 6 presents different tools for text to concept mapping in the medical domain and UMLS::Similarity module for computing semantic similarities on UMLS. These tools are essential to implement the proposed scenarios in corresponding platforms in order to carry out the experiments and test the different approaches in the medical domain.

2 Involving semantics in supervised text classification: a conceptual framework

According to the literature reviewed in previous chapter, many works proposed approaches involving semantics into the process of text classification at different steps of processing. Many works argued the utility of semantics at text representation step (Caropreso *et al.*, 2001; Liu *et al.*, 2004; Bloehdorn *et al.*, 2007; Séaghdha *et al.*, 2008; Wang *et al.*, 2008; Aseervatham *et al.*, 2009; Z. Li *et al.*, 2009; Séaghdha, 2009). Most of these works transformed the classical BOW into a BOC, choosing concepts as an alternative feature to words (Bloehdorn *et al.*, 2006; Hliaoutakis *et al.*, 2006; Mihalcea *et al.*, 2006; Gabrilovich *et al.*, 2007; Guisse *et al.*, 2009; Bai *et al.*, 2010; L. Huang *et al.*, 2012).

In addition, many state-of-the-art works deployed semantic similarity between concepts as well as concepts in text classification at two different steps: representation enrichment and prediction. Three major approaches are distinguished for representation enrichment: *Semantic Kernels* (usually deployed with SVM classifiers) (Bloehdorn *et al.*, 2007; Wang *et al.*, 2008; Aseervatham *et al.*, 2009; Séaghdha, 2009), *Generalization* (Bloehdorn *et al.*, 2006) and *Enriching Vectors* (L. Huang *et al.*, 2012). As for prediction step, only few works considered semantics in this step through *Text-To-Text Semantic Similarity Measures* that aggregate semantic similarity between concepts pair-to-pair (Hliaoutakis *et al.*, 2006; Mihalcea *et al.*, 2006; Guisse *et al.*, 2009; L. Huang *et al.*, 2012). Finally, some authors used the entire hierarchy or parts of it as a representation model, a classification model and a basis for prediction (Peng *et al.*, 2005; J. Z. Wang *et al.*, 2007; Guisse *et al.*, 2009).

In this work, we intend to investigate the previous approaches and apply them in the medical domain in order to assess their influence on a supervised text classification. We exclude two approaches from this investigation. The first approach is *Generalization* that is not suitable in a specific domain application, as adding superconcepts to the BOC introduces noise to the system and can deteriorate the classification accuracy. The second one is using the ontology as a representation and classification model, which is highly expensive especially when using large ontology.

This section presents a conceptual framework that summarizes all approaches considered in this work, aiming at involving semantics in the process of supervised text classification in the medical domain. Figure 28 illustrates a framework that involves semantics at the four following steps of the classification process.

First, we choose concepts as alternative feature to words in the classical vector space model. Thus, we involve semantic knowledge in indexing by using concept in text representation. **Conceptualization** is the process of finding a match or a relevant concept in a semantic resource that conveys the meaning of a word or multiple words from text. Concepts covering a text document constitute its semantic vector that can represent the document as a BOC in text classification or any other similar treatment. The main difficulty that faces a conceptualization process is ambiguous words. Usually, **disambiguation strategies** (Bloehdorn *et al.*, 2006) resolve such problems and identify matched concepts with the accurate meaning.

Second, we intend to investigate the impact of enriching text representation by means of *Semantic Kernels* (Wang *et al.*, 2008) that we apply on vectors representing the training corpus and the test documents after indexing. This enrichment is possible via a *Proximity Matrix*, which is built using semantic similarities between concepts of the BOC pair-to-pair. This BOC is the result of the previous conceptualization.

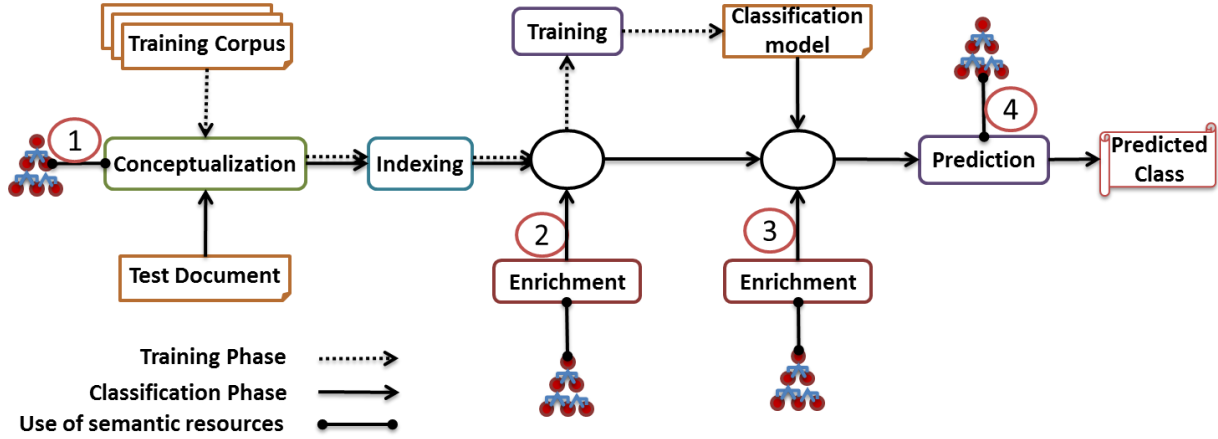


Figure 28 A conceptual framework to integrate semantics in supervised text classification process.

Third, we intend to investigate another approach for enriching BOC so called *Enriching Vectors* (L. Huang *et al.*, 2012). This approach enriches the classification model and test documents before prediction using proximity matrix as well.

Forth, we study and propose new *Text-To-Text Semantic Similarity Measures* that a classifier (like Rocchio) can use in class prediction. These measures deploy proximity matrix and aggregate semantic similarity between concepts of the compared vectors into semantic text-to-text similarity.

In fact, we are mainly interested in involving semantics in text classification in the medical domain. This is due to the difficulties faced by many researchers when classifying specific domain text documents (Bloehdorn *et al.*, 2006; Bai *et al.*, 2010), the fact demonstrated in previously presented results in chapter 2. Moreover, many researchers reported that using domain specific semantic resources for text classification in these domains improves its effectiveness (Bloehdorn *et al.*, 2006; Aseervatham *et al.*, 2009; Guisse *et al.*, 2009).

3 Involving semantics through text conceptualization

Involving semantics in text representation is by definition integrating concepts-as a unit of knowledge- into the indexing process. We refer to this integration by *Conceptualization* process. Most of the state of the art works apply conceptualization to vectors (Bloehdorn *et al.*, 2006; Ferretti *et al.*, 2008). We choose to apply conceptualization to raw text in order to take benefits of the syntactic and the semantics residing in text that text indexing generally ignores (YanJun Li *et al.*, 2008).

This section presents first different strategies for conceptualization and disambiguation. Then it presents a generic platform for text conceptualization.

3.1 Text Conceptualization Task

In the intent to overcome the limitations of word-based indexing, our framework uses semantic resources, such as thesauri and ontologies, to replace term-based representation by a concept-based one. Thus, a classification technique can deploy the semantically enriched presentation in classifying text.

Conceptualization is by definition “to interpret in a conceptual way” (“Cambridge Dictionaries Online, Cambridge University Press”, 2013). In the context of text analysis, it is the process of mapping literally occurring words in text to their corresponding concepts or senses as matched in semantic resources. Applying indexing to conceptualized text might improve classification results (YanJun Li *et al.*, 2008). According to the literature, conceptualization was applied to words using different strategies (Hotho *et al.*, 2003). As an example of semantic resources that were used for conceptualization: WordNet (Miller, 1995), Wikipedia (2013) and other domain specific resources usually called domain ontologies such as UMLS (2013) in the medical domain. In general, text conceptualization is realized through two steps:

- Analyze text in order to find candidate words for word to concept mapping.
- Search for corresponding concepts related to candidate words, and finally integrate these concepts in text producing the final conceptualized text.

Next subsection presents the different conceptualization strategies or the different ways to integrate the mapped concepts into the final conceptualized text. Then we present different strategies for facing ambiguities.

3.1.1 Text Conceptualization Strategies

During conceptualization, we map text words to their corresponding concepts in the semantic resource. Next step is to incorporate these concepts into the resulting text. According to the literature, three different strategies are possible to conceptualize word vectors (Bloehdorn *et al.*, 2006). We adapt these strategies to our approach for text conceptualization as the following:

- *Adding Concepts*: This strategy expands the original text using the mapped concepts. Conceptualized text contains original words as well as concepts (Ferretti *et al.*, 2008).

- *Partial Conceptualization*: This strategy, substitutes words by their corresponding concepts and keeps words having no related concepts in text. Conceptualized text contains mapped concepts and some original words (YanJun Li *et al.*, 2008).
- *Complete Conceptualization*: Similarly, to Partial Conceptualization, this strategy also substitutes words by concepts. The main difference is that it eliminates remaining words from the final conceptualized text that should contain concepts only (Bai *et al.*, 2010).

According to authors in (Bloehdorn *et al.*, 2006), the second strategy seems to be the most appropriate one as it replaces words by a related concept so no original feature is removed from the text (compared with the third one), and no extra feature is added (compared with the first one) resulting in minimized effects on efficiency. However, this is not the choice of authors in (Ferretti *et al.*, 2008) that used *Adding concepts* or authors in (Bai *et al.*, 2010) that used *Complete conceptualization*.

One of the objectives of this work is to study the effect of conceptualization on text classification using different conceptualization strategies. Yet, it is necessary to adapted indexing and classification techniques to hybrid text contents (concepts + words) and to investigate the effect of these strategies on classification as well. This is the main concern of the first part of the experimental study presented in next chapter.

3.1.2 Disambiguation Strategies

While searching the semantic resource for a mapping of a polysemous word, conceptualization may find multiple matches with different meanings. For example, the word "Book" signifies in English a book and a reservation (Ticket, accommodations, etc.). To face this problem, state of the art approaches for conceptualization proposes multiple strategies to deal with ambiguities (Bloehdorn *et al.*, 2006). Here we cite three strategies for disambiguation that can help solving this problem:

- *All*: this strategy accepts all candidate concepts as matches for the considered word.
- *First*: this strategy accepts the most frequently used concept among the different candidates according to language statistics.
- *Context*: this strategy accepts the candidate concepts having the most similar semantic context, as compared to the original word's context in the document (Aronson *et al.*, 2010; Bai *et al.*, 2010).

The first strategy, despite being the simplest, is the least reliable as it accepts all candidate concepts without choosing the appropriate sense of the word. The second strategy is more reliable. Nevertheless, this strategy fails to choose the right candidate concept if the correct sense corresponds to the rarely used sense of the word. Despite its complexity, the last strategy seems to be the most reliable and accurate (Bloehdorn *et al.*, 2006) and was deployed by most of state of the art approaches that treat ambiguities (Aronson *et al.*, 2010; Bai *et al.*, 2010). The context of a concept is related to its definition or its descriptive words in the semantic resource or to its textual context in a text corpus.

3.2 Generic framework for text conceptualization

Previous section presented different strategies for conceptualization and for resolving ambiguities during conceptualization. This section presents a generic framework for text conceptualization through different processing steps (see Figure 29). First step breaks up text into tokens and identifies candidate N-grams for concept matching. This step deploys different *Natural Language Processing* (NLP) techniques to analyze the text syntactically. Then, this framework searches for matches in the semantic resource for each of the candidates. These matches are lexically similar to the candidates or to their derivations. If the system finds multiple matches for the same candidate, it applies a disambiguation strategy to resolve this ambiguity in order to choose the correct match. Finally, the system integrates the matched concepts into the original text according to the conceptualization strategy in order to produce the final conceptualized text. We choose to apply conceptualization to raw text in the intent to implicate its syntactic and semantics in the process of conceptualization. This framework is generic and modular; different techniques and different application domains can fit in the system.

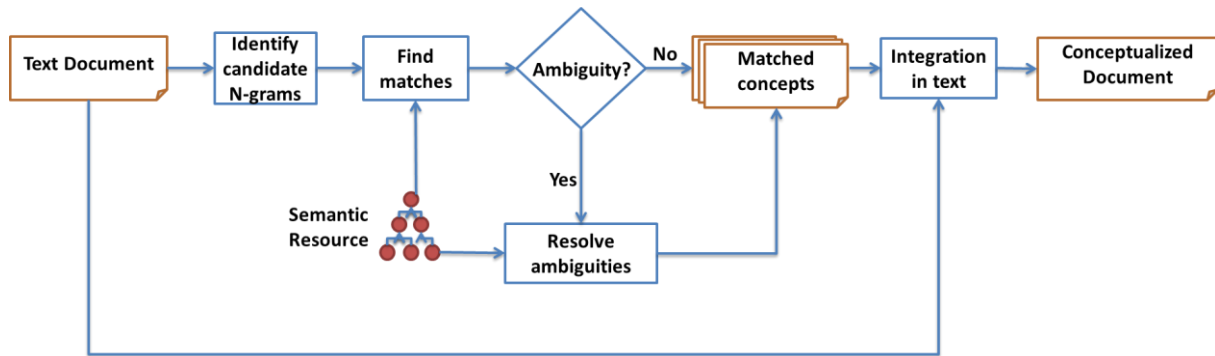


Figure 29. Generic platform for text conceptualization

3.3 Conclusion

In this section, we studied involving semantics in indexing through conceptualization. In the proposed approach, we apply conceptualization to text and enrich it with concepts in the ontology to which text is mapped. We discussed different strategies for conceptualization and for resolving ambiguities as well and presented a generic framework for text conceptualization. Contrarily to other approaches, we apply conceptualization on plain text in order to take advantage of its syntactic information and composed words. We intend to apply indexing on the conceptualized text using different conceptualization strategies and to test different text classification techniques. The main goal of this experimental study is to assess the influence of involving semantics in indexing on text classification. We will investigate these subjects in next chapter.

4 Involving semantic similarity in supervised text classification

Section 3 of this chapter presented a generic framework to transform the classical BOW into a BOC or a mix of both models according to the used conceptualization strategy. Using a complete conceptualization strategy, the result of conceptualization is a BOC constituted of ontology concepts to which text was mapped. Having a BOC model for conceptualized text classification; two further semantic integrations are feasible: enriching vectors with related concepts and assessing semantic text-to-text similarity. Both tasks use semantic similarity between concepts of the model vocabulary. In this aim, this section focus is on semantic similarity and proximity matrix that are used in enriching the BOC with similar concepts and in assessing the semantic similarity at document level for class prediction as well.

This section presents first a summary on semantic similarity measures. Then it presents a generic framework for building proximity matrices using engines for assessing semantic similarity between concepts on an ontology. The product of this framework, which is the proximity matrix, is essential to enriching BOC with similar concepts using either *Semantic Kernels* or *Enriching Vectors*. Finally, this section presents using semantic similarity in class prediction through new *Text-To-Text Semantic Similarity Measures*.

In this study, we will apply conceptualization to different classification techniques contrarily to semantic enrichment and prediction that we will apply to Rocchio classifier. Our choice for Rocchio as the classification technique to test the last two tasks is due to its extensibility for semantic integration not only by enriching document representation but also by enriching the classification model. What makes of Rocchio a special case is the fact that its classification model is composed of vectors at the center of the spheres delimited by the training documents of each class. In fact, these vectors are also BOC if built on the BOCs of the training documents, and so we can enrich them by means of either of the two representation enrichment techniques. Moreover, Rocchio uses similarity measures as the prediction criterion, these measures can be replaced by Semantic Text-to-Text similarity measures when using BOC for text representation.

4.1 Semantic similarity

In previous chapter, we reviewed state of the art semantic similarity measures and identified three major families: Ontology-based measures, Information theoretic-based measures and Feature-based measures. The fourth family is Hybrid measures that combine multiple principles from different families.

We compared different measures from these families and concluded that the most attractive family is Ontology-based measures as it depends only on the structure of the ontology. Its simplicity is the origin of its demonstrated efficiency in different application domains where semantic similarity is required and deployed (Sanchez *et al.*, 2012). Moreover, many authors argue that ontology is an explicit model of the knowledge in the domain it represents, and deploying this knowledge is sufficient to assess semantic similarities among its concepts (Seco *et al.*, 2004; Pirro, 2009). In fact, most ontologies produced by research projects

are fine grained and consistent so they fulfill the conditions of ontology-based measures. In other words, using such ontologies can guarantee effective and efficient ontology-based semantic similarity measures.

In this work, we are mainly interested in semantics in the medical domain. We intend to use the UMLS® as the semantic resource for assessing similarities in the semantic similarity engine (see Figure 30).

4.2 Proximity matrix

As mentioned earlier, semantic similarity is used in involving semantics through enriching representation and assessing semantic similarity between documents. Using an ontology, we can assess semantic similarity between the concepts of the vocabulary of the BOC model pair-to-pair. We propose to constitute a proximity matrix using these similarities.

Proximity matrix (PM) is a square matrix in which each cell (i,j) is a measure of similarity (or distance) between elements to which row (i) and column (j) correspond. Using a symmetric similarity measure, resulting proximity matrix is symmetric and vice versa.

Figure 30 illustrates a framework to build a proximity matrix for a vocabulary covering the features of a BOC model. In fact, indexing a corpus of text documents, after a complete conceptualization, results in a vocabulary of (n) concepts. Given the resulting vocabulary, a semantic similarity engine can constitute a proximity matrix by means of a semantic similarity tool. Thus, the semantic similarity tool assesses the semantic similarity between each pair of concepts from the vocabulary (c_i, c_j) and the engine assigns it to the corresponding cell in the proximity matrix.

In general, calculating semantic similarities between concepts of a semantic resource is a time consuming task and can affect the efficiency of the semantic platform in which it is integrated. This drawback is due to many factors like the size and the coverage of the semantic resource and the complexity of the chosen semantic similarity measure. Furthermore, this deterioration in efficiency depends also on the semantic platform itself and on the specific task that requires calculating proximity matrices or semantic similarities. The intensive use of the semantic similarity engine in a semantic platform results in significant deterioration in efficiency.

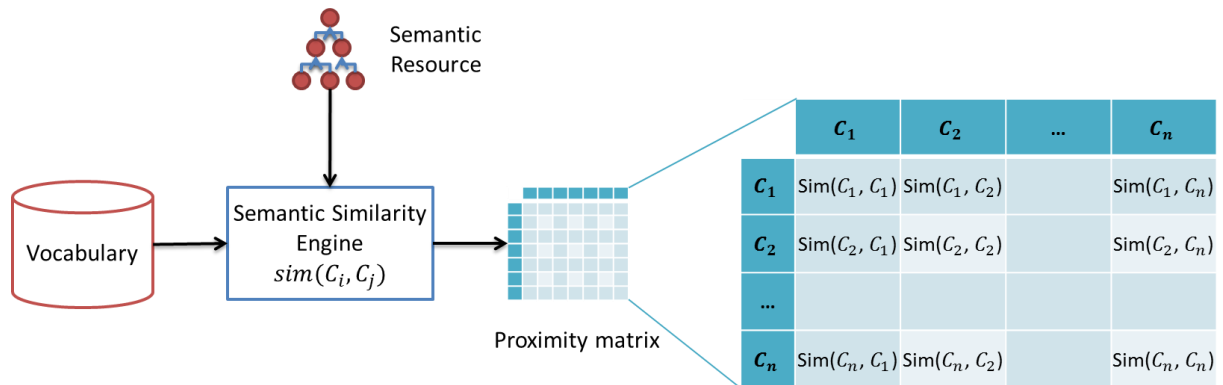


Figure 30. Building proximity matrix for a vocabulary of concepts of size n .

4.3 Semantic kernels

In general, semantic kernels are used with SVM (Bloehdorn *et al.*, 2007; Wang *et al.*, 2008; Aseervatham *et al.*, 2009; Séaghdha, 2009) in order to transform the original BOC into a new one in which training examples are linearly separable. Many state of the art approaches deployed general-purpose semantic resources in building their semantic kernels like WordNet (Bloehdorn *et al.*, 2007; Séaghdha, 2009) and Wikipedia (Wang *et al.*, 2008). Others used domain specific ontologies like UMLS for the medical domain (Aseervatham *et al.*, 2009).

Authors in (Wang *et al.*, 2008) made decisions for efficiency and limited the number of related concepts used in enrichment. Having a BOC model composed of N concepts, authors chose the five most similar concepts to those that constitute the model. The weight assigned to an added concept is the sum of the products of weights of each related concept and its semantic similarity to the added concept.

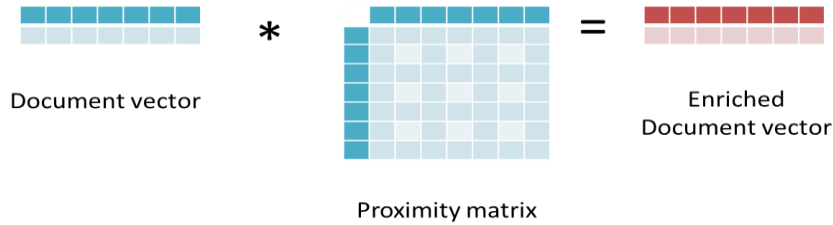


Figure 31. Applying semantic kernel to a document vector

In order to enrich a document representation using a semantic kernel, we need the BOC representing this document and a proximity matrix built for the N concepts of this BOC using a semantic similarity measure. In addition, one can limit the number of related concepts used in the semantic kernel to the k most similar concepts. We propose to apply the semantic kernel method for enriching vectors according to the following steps (see Figure 31):

1. Limit to the k most similar concepts of each concept in the vocabulary:
For each concept i of the vocabulary:
 - a. Identify the most similar concepts in the i th column of the proximity matrix
 - b. Set the cells corresponding to other concepts in the proximity matrix to 0
2. Apply the semantic kernel to each document d_i
 - a. Get the vector representing the document d_i using BOC model
 - b. Calculate the product $d_i * PM'$

PM' is the proximity matrix after limiting the number of related concepts to use in the kernel according to the first step. We formalize the previous steps in the following algorithm:

```

FOR each column in the proximity_matrix
  CALL MaxSimilarConcepts with proximity_matrix[column], k RETURNING
  MaxSim[k]
  FOR each row in the proximity_matrix
    IF NOT proximity_matrix position (row, column) in MaxSim[k]
      THEN SET proximity_matrix position (row, column) to ZERO
    END IF
  END FOR
END FOR
FOR each document in the corpus
  Get document_vector
  CALCULATE matrix product of document_vector, proximity_matrix
END FOR

```

Figure 32 illustrates how to apply the semantic kernel on a conceptualized document (using a complete conceptualization strategy). First, indexing builds the vector representing the text document as a BOC. Then, the system applies the semantic kernel method, using a proximity matrix, to the vector in order to enrich text representation with similar concepts. After applying semantic kernel to the vectors representing documents in the BOC model, resulting vectors are in general less sparse which might help Rocchio learn the classification model and predict classes of new documents.

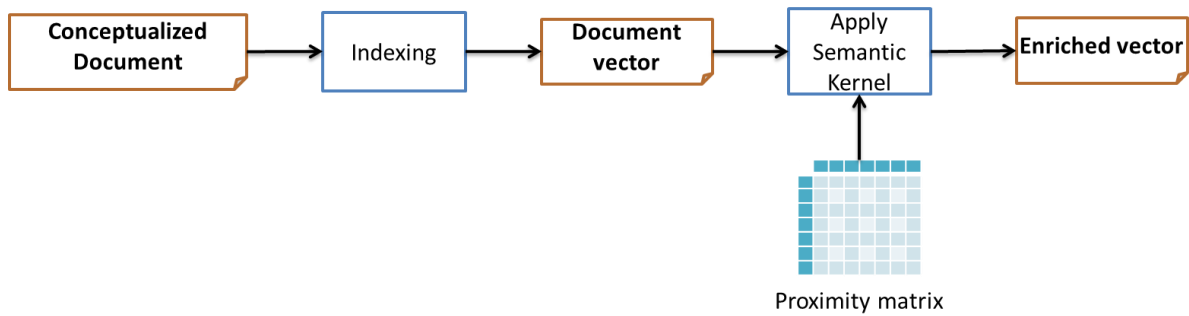


Figure 32. Steps to apply semantic kernel to a conceptualized text document

4.4 Enriching vectors

Authors in (L. Huang *et al.*, 2012) proposed this method and applied it in the context of text clustering using K-means, and in text classification using K Nearest Neighbors (KNN). In order to compare two documents, authors apply this method to the vectors that represent these documents and then apply a classical text-to-text similarity measure like Cosine. This method demonstrated a better correlation with human judgment as compared to applying the classical similarity measure on the original vectors.

Classical similarity measures, that we usually deploy to compare text documents represented in the vector space model like Cosine, depend on lexical matching in text comparison. In fact, these measures take into consideration the shared features among the compared vectors neglecting any other similarities such as semantic similarity among the unshared features. In other words, if two texts do not share the same words but use synonyms, they are presumed dissimilar. We previously identified this drawback of the classical BOW.

In order to go beyond lexical matching, we intend to apply Enriching vectors to each pair of vectors before comparison. By means of this method, each of the compared vectors enriches the other vector using its exclusive features. As a result, the vectors become less sparse, which makes applying the classical similarity measures more effective.

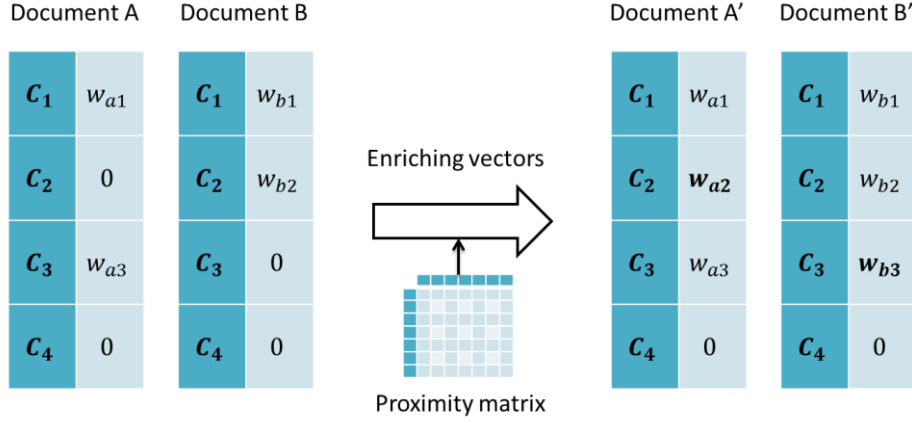


Figure 33. Applying Enriching vectors to a pair of documents. As a result, the weight corresponding to C_2 in A changes from 0 to w_{a2} and the weight corresponding to C_3 in B changes from 0 to w_{b3} . The vocabulary size is limited to 4.

To have a close look on the approach, Figure 33 illustrates how it works on a pair of documents. Given two documents A, B represented using a vocabulary of four concepts, we note that C_2 is an exclusive feature for B (mapped to B's text only) and that C_3 is an exclusive feature for A. The main goal of this approach is to give C_2 an appropriate weight in A and C_3 an appropriate weight in B. These weights are estimated using weights of other features of the document and the semantic similarities between these features and the missing feature according to the following formulas:

$$w(c, A) = w(SC(c, A)) * sim(c, SC(c, A)) * CC(c, A) \quad (70)$$

Where:

$w(SC(c, A))$ is the weight of the Strongest Connection (SC) of the concept c in A which is the weight of the most similar concept in A

$sim(c, SC(c, A))$ is the similarity between the concept and its strongest connection

$CC(c, A)$ is the Context Centrality (CC) of the concept c in the document A that is given by the following formula:

$$CC(c, A) = \frac{\sum_{c_i \in A} sim(c, c_i) * w(c_i, A)}{\sum_{c_i \in A} w(c_i, A)} \quad (71)$$

Where:

$sim(c, c_i)$ is the similarity between the concept c and the concept c_i from the document A.

$w(c_i, A)$ is the weight of concept c_i in the document A.

Assuming that C_3 is more similar to C_2 than C_1 , the following formula calculates the weight of C_2 in A' after enrichment:

$$w_{a2} = w_{a3} * \text{sim}(C_2, C_3) * \frac{\sum_{C_i \in A} \text{sim}(C_2, C_i) * w_{ai}}{\sum_{C_i \in A} w_{ai}}$$

Note that a classical similarity measure identifies one common feature between the vectors A, B before enrichment which is (C_1), and three features after enrichment(C_1, C_2, C_3). Therefore, the assessed similarity on the original vectors is different from the one assessed on vectors after enrichment.

Having two documents A and B represented as a BOC, here are the steps for enriching the vector mutually:

1. Search the vectors A, B for an exclusive feature C_i
2. If C_i is in A and not in B
 - a. Search in B for the most similar feature C_j (having non zero weight)
 - b. Calculate the weight w_{bi} and assign it to the feature C_i in B
3. Else (in B and not in A)
 - a. Search in A for the most similar feature C_j (having non zero weight)
 - b. Calculate the weight w_{bi} and assign it to the feature C_i in A
4. Repeat 1 until the vocabulary is covered

We formalize the previous steps in the following algorithm:

```

FOR each document pair
  FOR each feature i in the vocabulary
    If A position(i) != 0 AND B position(i) = 0 THEN
      CALL findMaxSim WITH B and i AND PM RETURNING j
      CALCULATE weight_iB WITH weight_jB and B and PM
      SET B position(i) to weight_iB
    ELSE
      IF B position(i) != 0 AND A position(i) = 0 THEN
        CALL findMaxSim WITH A and i AND PM RETURNING j
        CALCULATE weight_iA WITH weight_jA and A and PM
        SET A position(i) to weight_iA
      END IF
    END IF
  END FOR
END FOR

```

PM is the proximity matrix that stores the semantic similarity between the concepts of the feature space pair-to-pair. The function `findMaxSim` searches a vector for the most similar feature to a specific feature (passed as a parameter) and a non-zero weight.

Figure 34 illustrates the different steps to apply Enriching vectors on two text documents that are conceptualized using a complete conceptualization strategy. First, indexing step extracts conceptual features from the documents and transform them to vectors as BOCs. By means of a proximity matrix (using a particular semantic similarity measure), both vectors are mutually enriched as a second step. Finally, we compare the enriched vectors using a classical similarity measure. The resulting similarity takes into consideration similar concepts as well as common concepts.

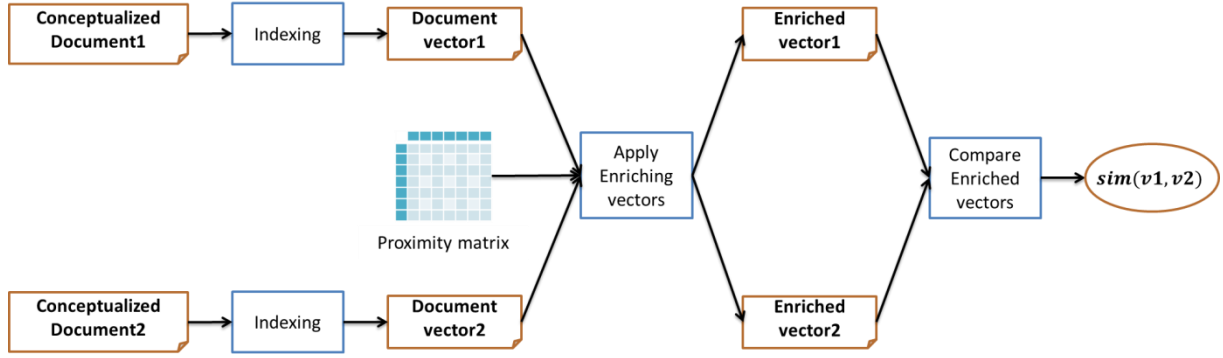


Figure 34. Steps to apply Enriching vectors to a pair of conceptualized text documents

This approach is compatible with text classification using Rocchio, its application is straightforward by replacing document vector1 with the vector of a centroid, and the document vector2 with the vector of the conceptualized document. Thus, vectors representing the centroid and the document are enriched mutually before their comparison by means of a similarity measure between them. Experimental study will assess the influence of this approach on the effectiveness of Rocchio in next chapter.

4.5 Semantic measures for text-to-text similarity

Previous subsections discussed approaches for involving semantics in indexing and in learning the classification model. In general, most research on semantic similarity concerns semantic similarity between concepts of ontologies pair-to-pair. Semantic similarity at document level is rarely investigated.

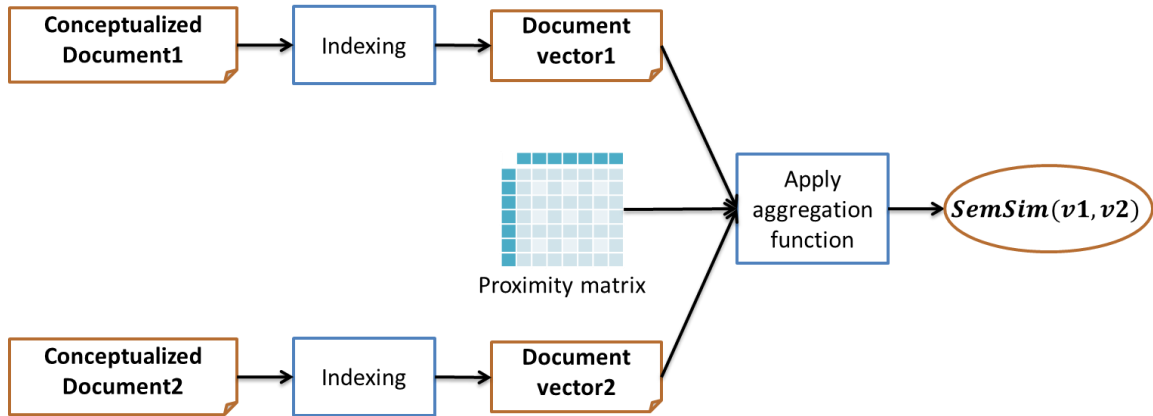


Figure 35. Steps to applying aggregation function on a pair of conceptualized documents

In this subsection, we are interested in involving semantics in new *Text-To-Text Semantic Similarity Measures*. Some classifiers like Rocchio use this kind of measures in class prediction as the criterion with which they choose the most similar class for a treated document. In this work, we will deploy some of the state of the art measures and propose a new measure for assessing the semantic similarity between two BOCs representing two text documents (or a document and a centroid in the case of Rocchio). These measures are functions that aggregate semantic similarities between concepts of the compared documents pair-to-pair. We apply complete conceptualization to both documents, and then indexing represents them as BOCs.

Finally, an aggregation function calculates the semantic similarity between the documents using their representation and the semantic similarities between their concepts pair-to-pair that are stored in the proximity matrix. Figure 35 illustrates different steps for applying aggregation functions to a pair of documents.

Rada *et al.* (1989) proposed the first aggregation function that calculates the semantic similarity between two groups of concepts using the mean of all combinations of pairs of concepts between these groups using the following formula:

$$Sim_g(g_1, g_2) = \frac{1}{m * n} \sum_{i \in g_1} \sum_{j \in g_2}^n Sim(c_i, c_j) \quad (72)$$

Where:

m, n are the number of concepts in g_1, g_2 respectively

$sim(c_i, c_j)$ is the semantic similarity between the concepts c_i from g_1 and c_j from g_2

Azuaje *et al.* (2005) proposed a similar aggregation function that takes into consideration maximum semantic similarities between each concept of g_1 and all concepts from g_2 and vice versa according to the following formula:

$$Sim_g(g_1, g_2) = \frac{1}{m + n} \left(\sum_{c_i \in g_1}^m \max_{c_j} (Sim(c_i, c_j)) + \sum_{c_j \in g_2}^n \max_{c_i} (Sim(c_i, c_j)) \right) \quad (73)$$

In fact, previous fomulas are adequate to compare two groups of concepts where all concepts have equal importance to the system. Nevertheless, in the context of text classification or information retrieval, each concept is assigned an importance according to its occurrence frequency by means of a weighting scheme. In the intent of adapting the previous measure to the context of information retrieval, Hliaoutakis *et al.* (2006) proposed the following semantic similarity measure for ranking MEDLINE document d according to a particular query q where both are presented as BOCs:

$$Sim(q, d) = \frac{\sum_i \sum_j q_i d_j sim(c_i, c_j)}{\sum_i \sum_j q_i d_j} \quad (74)$$

Where:

q_i, d_j are the weights of concept i in the query and the concept j in a document

$sim(c_i, c_j)$ is the similarity between the concept c_i from the query q and c_j from the document d .

Similarly to the previous approach, Mihalcea *et al.* (2006) proposed a new aggregation function to compare short texts or phrases. In fact, this function combines the two previous approaches as it takes into consideration pairs of concepts having maximal similarities and the corresponding Inverse Document Frequency (IDF) as well. The aggregation function is calculated following this formula:

$$SemSim(T_1, T_2) = \frac{1}{2} \left(\frac{\sum_{w \in T_1} \max Sim(w, T_2) * idf(w)}{\sum_{w \in T_1} idf(w)} + \frac{\sum_{w \in T_2} \max Sim(w, T_1) * idf(w)}{\sum_{w \in T_2} idf(w)} \right) \quad (75)$$

Where:

$\max Sim(w, T_2)$ is the maximum similarity between word (w) and all words in (T_2)

$idf(w)$ is the inverse document frequency of the word (w)

Previous aggregation functions are used to assess semantic similarity between two text documents or two phrases or in ranking documents for a particular query. In this study, we are particularly interested in text classification. Among the classification techniques we used so far, Rocchio is the only classifier that deploys similarity measures for class prediction like Cosine, Jaccard and so on. In other words, Rocchio is the only classifier that accepts involving semantics in class prediction.

In this work, we propose a new aggregation function (*AvgMaxAssymTFIDF*) adapting the previous one to text classification by using TFIDF weights instead of IDF weights in order to take into consideration the importance of a word in a document instead of its importance in the corpus. This function becomes as the following formula:

$$SemSim(T_1, T_2) = \frac{1}{2} \left(\frac{\sum_{w \in T_1} maxSim(w, T_2) * TfIdf(w)}{\sum_{w \in T_1} TfIdf(w)} + \frac{\sum_{w \in T_2} maxSim(w, T_1) * TfIdf(w)}{\sum_{w \in T_2} TfIdf(w)} \right) \quad (76)$$

Where:

$maxSim(w, T_2)$ is the maximum similarity between word (w) and all words in (T_2)

$TfIdf(w)$ is the normalized frequency of the word (w) according to the TFIDF weighting scheme.

In next chapter, we will test Rocchio replacing classical similarity measures by semantic similarity measures using one of the previous aggregation functions. We will investigate their influence on Rocchio's effectiveness.

4.6 Conclusion

Using the BOC model that represents a completely conceptualized text, this section presented approaches involving semantic similarities in supervised text classification by enriching text representation and semantic class prediction. All of these approaches deploy semantic similarities between concepts of the BOC in the form of a proximity matrix.

In this aim, this section presented a summary on semantic similarities and a generic framework that generates proximity matrix. The proximity matrix built on the vocabulary of the BOC model is the major component in the three proposed approaches.

This section presented two ways of enriching BOC using related concepts in the ontology: semantic kernels and enriching vectors. Both techniques intend to overcome the limitations of classical similarity measures that are usually based on lexical matching ignoring the semantics the features convey. By enriching vectors with similar concepts, the comparison between the resulting vectors becomes more effective using classical similarity measures.

The third approach presented in this section involves semantic similarity in classification through aggregation functions that can be used for prediction. Aggregation functions aggregate semantic similarities between concepts of the vocabulary pair-to-pair in a semantic text-to-text similarity measure. This measure is then used in comparing vectors in the feature space. We proposed a new aggregation function that will be tested in next chapter.

In this study, we will apply the three proposed approaches of this section to Rocchio classifier that accepts semantic integration; Rocchio's classification model or centroids are

vectors that contain all the features of the training documents. Thus, its classification model accepts semantic enrichment and its prediction process accepts involving semantics through semantic text-to-text similarity measure.

Next section presents our methodology and the four scenarios involving semantics in supervised text classification that we implemented and tested in the medical domain in next chapter.

5 Methodology

Previous sections presented the approaches integrating semantics in the process of supervised text classification as illustrated in Figure 29. This section is focused on the methodology we followed in order to implement and evaluate these approaches. Here we propose a generic framework for each of the following four scenarios: *Conceptualization*, enrichment using *Semantic Kernels*, enrichment using *Enriching Vectors* and using *Semantic Text-To-Text Similarity Measures* in class prediction.

5.1 Scenario 1: Conceptualization only

This scenario follows the steps illustrated in Figure 36 and the specifications in section 3 in order to involve concepts in supervised text classification. This framework is very similar to a classical supervised text classification system after adding the conceptualization step before indexing. This conceptualization enriches text using appropriate concepts that are retrieved from the semantic resource. Conceptualized training corpus is indexed and handed over to the classification technique for training, whereas conceptualized test documents are indexed and then handed over to the classification technique for class prediction.

Concerning the conceptualization step, it implements specifications from section 3 including a conceptualization strategy and a disambiguation strategy following the generic schema represented in Figure 29. In this scenario, the role of semantics is limited to conceptualization whereas the rest of the framework is similar to a classical supervised text classification.

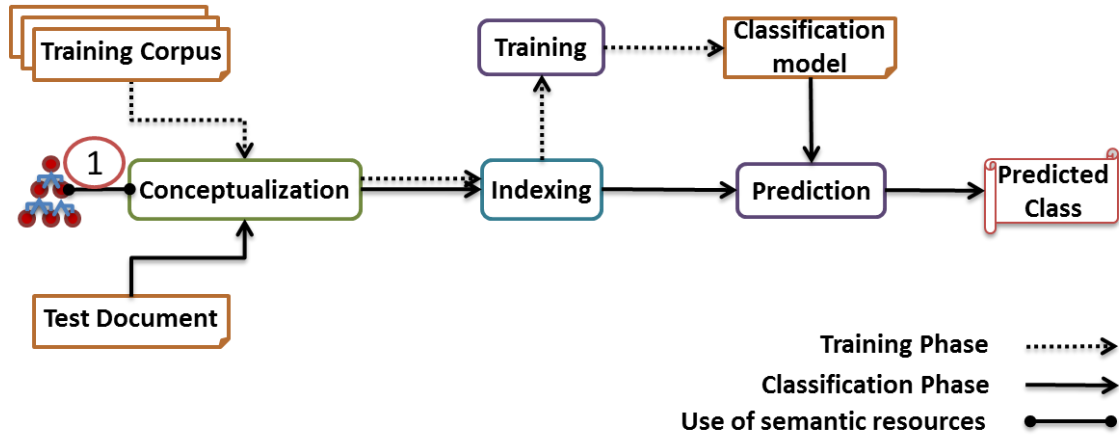


Figure 36. Generic framework for using text conceptualization in supervised text classification

5.2 Scenario 2: Conceptualization and enrichment before training

In this scenario, text classification deploys concepts and semantic similarities through conceptualization and enrichment steps correspondingly (see Figure 37). In this case, we use the complete conceptualization strategy in order to generate a BOC corresponding to text contents, and then we apply semantic kernels using the proximity matrix built on the vocabulary of the BOC model and the semantic resource using the specifications in section 4.3.

Similar to the previous scenario, this scenario applies complete conceptualization before indexing. Then, it enriches index of training documents before training. On the other side, it enriches index of test documents and hands it over to the classification technique in order to predict their classes using the learned model. The main difference between this scenario and the previous one is involving similar concepts in addition to those detected in text through conceptualization. Here, the framework deploys concepts and semantic relations between them in the semantic resource.

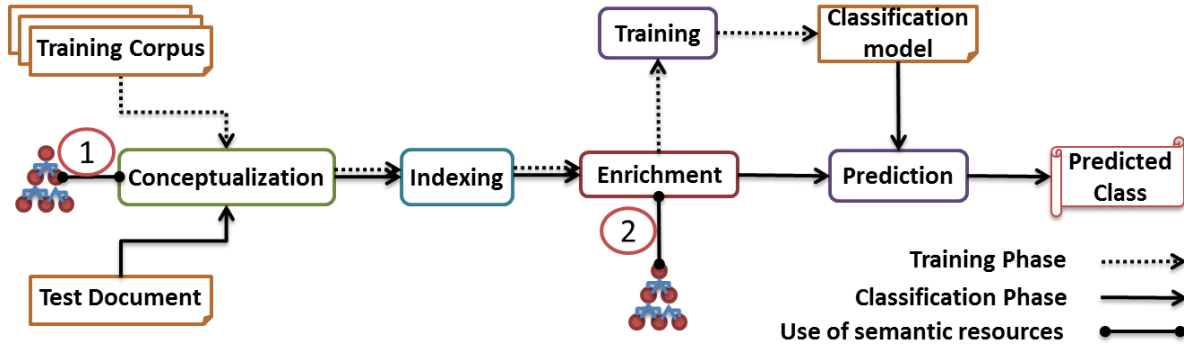


Figure 37. Generic framework using semantic kernels to enrich text representation

5.3 Scenario 3: Conceptualization and enrichment before prediction

In this scenario, text classification deploys concepts and semantic similarities through conceptualization and enrichment steps correspondingly (see Figure 38). In fact, this scenario is quite similar to the previous one except for the timing of enrichment; classification model and text document are mutually enriched just before prediction. In this case, we use the complete conceptualization strategy in order to generate a BOC and we apply Enriching vectors using proximity matrix that is built using the vocabulary of the model and the semantic resource using specifications in section 4.4.

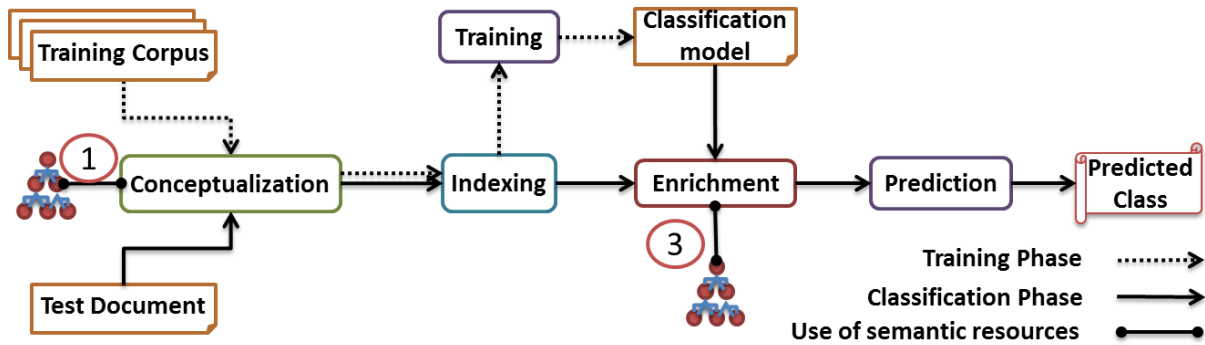


Figure 38. Generic framework using Enriching vectors to enrich text representation

This scenario, as the previous scenario, applies complete conceptualization before indexing. Then, it trains the classification technique on the index of training documents in order to build the classification model. On the other side, it indexes test documents and hands them over along with the classification model to enrich them mutually. Finally, it delivers enriched

indexes to the classification technique in order to predict their classes. As the previous scenario, this scenario deploys concepts and semantic relations between them from the semantic resource.

5.4 Scenario 4: Conceptualization and semantic text-to-text similarity for prediction

The fourth scenario is similar to the first one except for the use of semantics in class prediction. A generic framework for this scenario is illustrated in Figure 39. First, this framework uses a complete conceptualization strategy on input (training corpus and test documents) before indexing in order to generate a BOC. The rest of the framework is similar to a classical supervised text classification except for prediction that involves semantic resources according to specifications in 4.5. In this case, we apply semantic text-to-text similarity measures using a proximity matrix and an aggregation function.

As the previous two scenarios, this scenario deploys concepts and semantic relations between them in the semantic resource. Concepts are involved in text through conceptualization and relations are deployed to assess semantic similarities between concepts in order to estimate the semantic similarity between two groups of concepts representing the document and the classification model.

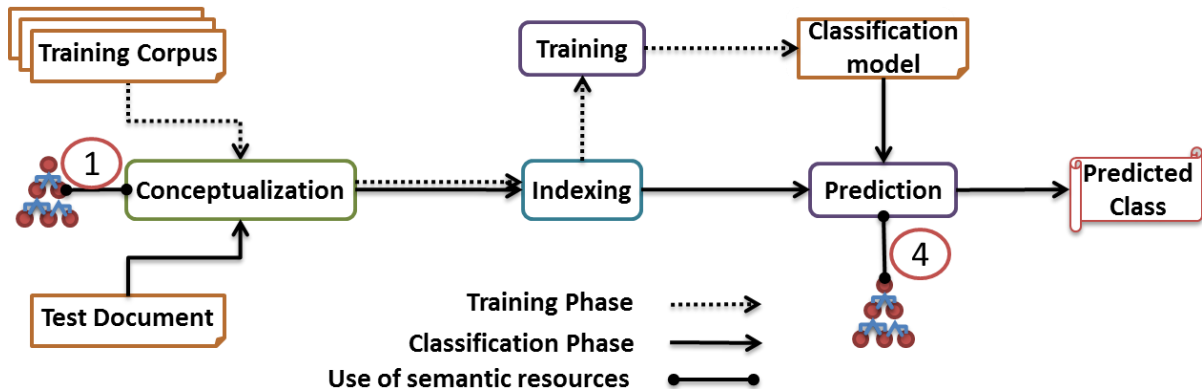


Figure 39. Generic framework for using semantic text-to-text similarity in class prediction

5.5 Conclusion

This section presented the methodology we used in investigating the role of semantics in supervised text classification. This methodology is applied through four scenarios *Conceptualization*, enrichment using *Semantic Kernels*, enrichment using *Enriching Vectors* and using *Semantic Text-To-Text Similarity Measures* in class prediction. The first scenario involves concepts only whereas the three other ones involve concepts as well as relations between them in the classification process. Furthermore, the first scenario is the minimal one using *Conceptualization* only, whereas all of the three other scenarios use *Conceptualization* with one of the three approaches involving semantic similarities. Note that the second scenario applies representation enrichment before training whereas the third scenario applies enrichment after training and before prediction. This section presented also a generic framework for each of the four scenarios that implement specifications of each of the deployed approaches.

Next section focuses on tools and technical details related to the medical domain that are necessary for the implementation of each of the presented scenario and for experimental study in next chapter.

6 Related tools in the medical domain

Previous sections presented specifications and scenarios for involving semantics in supervised text classification. This section provides details on tools that are related to the application domain which is the medical domain. These tools are essential to implement the previous scenarios. This section provides also some technical choices. First, this section presents tools for text to concept mapping and then it presents tools for semantic similarity all developed for the medical domain.

6.1 Tools for text to concept mapping

In general, probability distribution underlying medical texts is different from the distribution underlying texts in other domains (ASCH, 2012). Thus, specific semantic resources and adapted tools are necessary for better medical text treatment. In this section, we are interested in the well-known UMLS as a semantic resource in the medical domain and four tools for mapping medical text to concepts in UMLS.

This section presents four well known tools for text to concept mapping in the medical domain: PubMed Automatic Term Mapping ("PubMed Tutorial," 2013), MaxMatcher (X. Zhou *et al.*, 2006), MGREP (Shah *et al.*, 2009), and MetaMap (Aronson *et al.*, 2010). We introduce the process of mapping in each of these tools. Each tool has its own advantages and limitations, which are evaluated by means of their precision, recall and efficiency in mapping. We choose to use MetaMap for text to UMLS concept mapping in our system.

6.1.1 PubMed Automatic Term Mapping (ATM)

PubMed ("PubMed Tutorial," 2013) deploys this tool to find a match to user's search keywords or query phrase. PubMed ATM matches the phrase against subjects or concepts in multiple databases: MeSH (Medical Subject Headings) translation table, journals translation table, full author translation table, author index, full investigator translation table and investigator index. If it finds a match, the mapping stops. Otherwise, it breaks apart the phrase and repeats the process until a match is found. In addition, PubMed ATM searches the phrases and individual terms in All Fields. When matching text against concepts in MeSH, ATM searches for exact matches in MeSH subheadings, MeSH Synonyms, mappings derived from UMLS® and Supplementary Concepts.

Given the query "gene replication", PubMed ATM treats the text and tags matched terms using [MeSH Terms] and others with [All Fields]. Thus, the initial query is transformed into a boolean expression for search as the following: ("genes"[MeSH Terms] OR "genes"[All Fields] OR "gene"[All Fields]) AND ("dna replication"[MeSH Terms] OR ("dna"[All Fields] AND "replication"[All Fields]) OR "dna replication"[All Fields] OR "replication"[All Fields]).

6.1.2 MaxMatcher

Exact dictionary lookup, like PubMed ATM, has a well-known drawback: it searches exact matches in MeSH, which cannot treat all variations of medical terms. This results in low mapping recall (X. Zhou *et al.*, 2006). To overcome this limitation, authors in (X. Zhou *et al.*,

2006) proposed an *approximate dictionary lookup* technique for matching text to concepts. This technique matches text looking for its significant words in concepts rather than matching all words in concepts.

The major advantage of MaxMatcher (X. Zhou *et al.*, 2006) is that the order of words and inserting or deleting insignificant words into text does not affect the recognition of concepts. In fact, MaxMatcher gives matches scores that take into consideration the previous differences between the matched text and the matched concept name. In case of ambiguity, MaxMatcher considers the surrounding words in text in order to choose the best matching concept.

6.1.3 MGREP

The *National Center for Biomedical Ontology* (NCBO) developed a BioPortal (2013) for automated, ontology-based access to online medical resources. In this context, the NCBO adopted the University of Michigan's MGREP (Bhatia *et al.*, 2008; Dai, 2008) server for concept recognition in medical text.

Compared to previous tools, MGREP matching is more effective as it processes concepts in the knowledge base according to the procedures illustrated in Figure 40. After removing common words from concepts, MGREP applies all possible variations to each word and builds a tree using different word orders. When matching text to concepts, MGREP matches text to all concepts variations using a radix-tree match. Instead of using time-consuming complex approaches to generate concept variations during matching, MGREP generates concepts variations *a priori*, which makes matching more efficient.

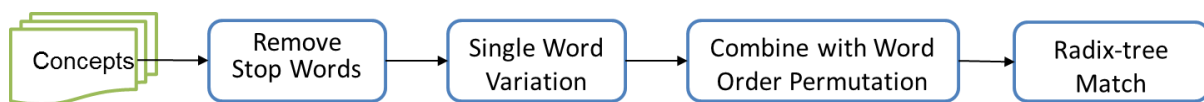


Figure 40. Concept processing in MGREP (Dai, 2008)

6.1.4 MetaMap

The major goal of MetaMap (Aronson *et al.*, 2010) developers at the NLM was to improve medical text retrieval using UMLS Metathesaurus. Indeed, MetaMap can discover links between medical text and the knowledge in the Metathesaurus.

Text to concept mapping in MetaMap is the result of a rigorous linguistic analysis of each phrase of the text (Aronson, 2001; Bashyam *et al.*, 2007; Aronson *et al.*, 2010) (Figure 41): First the text is tokenized and phrase boundaries are identified, then part-of-speech tags are added. Second, the Specialist lexicon and the shallow parser perform lexical lookup and syntactic analysis successively, which generates variants for the treated phrases. Finally, MetaMap identifies different candidates in the Metathesaurus and then combines them to generate final mappings to which it assigns confidence scores. Given the phrase “*Patients with hearing loss*” as an input to MetaMap, it divides the phrase into two parts: “Patients” and “with hearing loss”. Then, it treats each part separately and returns results in a ranked list of candidates. Taking the second part as an example, the candidate “hearing loss” had the score

1000 whereas the candidate “Hearing” had the score 861. MetaMap selects best candidates as mappings for each part of the input text. In our example, the mappings are “Patient”, “hearing loss” (partial hearing loss) and “hearing loss” (hearing impairment).

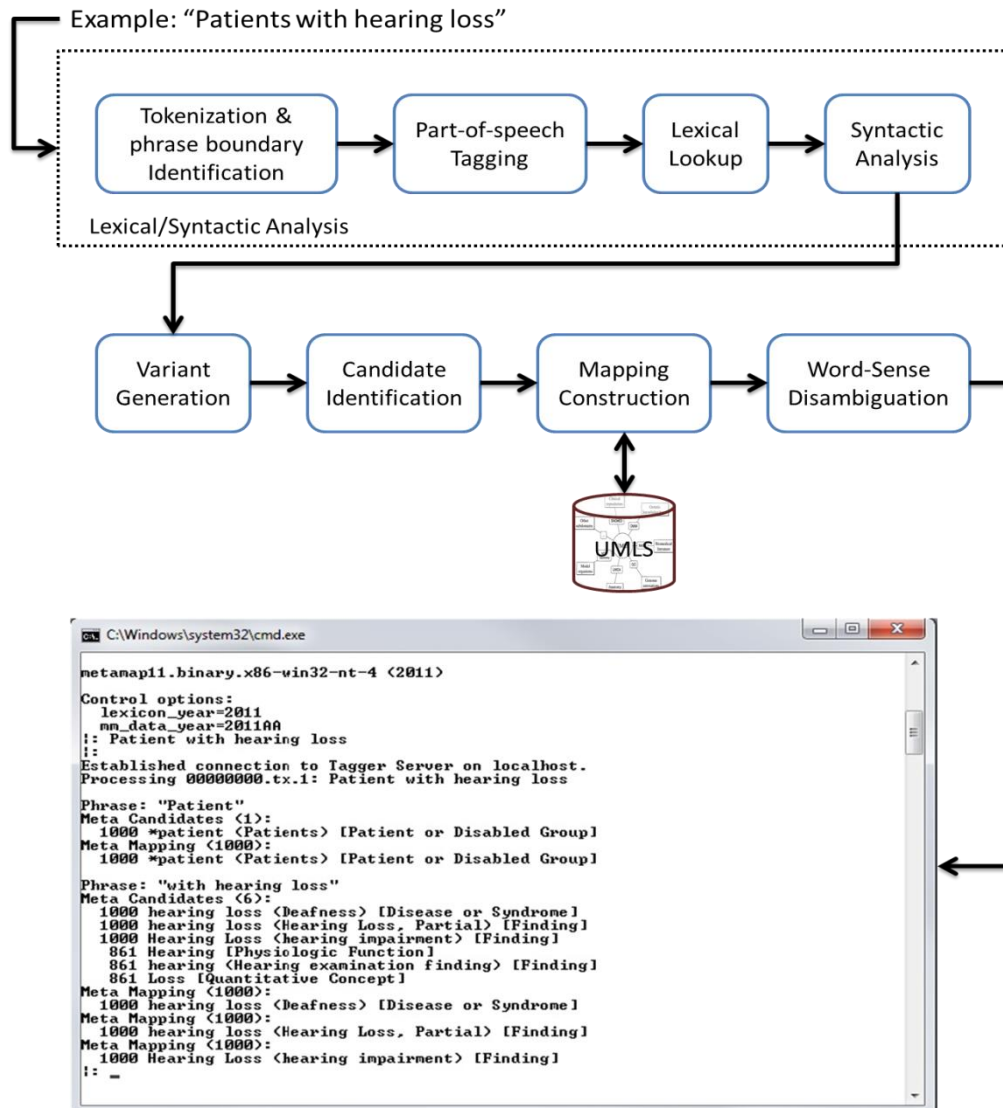


Figure 41. MetaMap: Steps for text to concept mapping (Aronson *et al.*, 2010). The example of command line output of MetaMap occurred using the phrase “Patients with hearing loss”.

In cases where MetaMap matches ambiguous words to different mappings, MetaMap keeps the most semantically similar mappings to the surrounding text following the context strategy (section 3.1.2).

MetaMap is an effective text to concept mapping tool according to many evaluations on different corpora. Thus, many applications in the medical domain deployed MetaMap like the *Medical Text Indexer* (MTI) (Aronson *et al.*, 2004) for indexing PubMed articles using MeSH concepts.

6.2 Tools for semantic similarity

This section concerns tools for assessing semantic similarity between concepts in the medical domain and the module that generates the proximity matrix. First, we detail the semantic similarity engine by which the proximity matrix is built. Then we introduce the module UMLS::Similarity and justify our choice of ontology-based semantic similarity measures.

6.2.1 Semantic similarity engine

The main module in the semantic similarity engine is for semantic similarity. Research projects developed many libraries to calculate semantic similarity between concepts of ontologies implementing different state of the art semantic similarity measures. We mention particularly SimPack, SML, SIMILAR, WordNet::Similarity, and UMLS::Similarity.

SimPack (Ziegler *et al.*, 2006), an open source java library, is a result of research on similarity between concepts in ontologies or between ontologies as a whole. SimPack is also suitable in other application domains like assessing the similarity between software source code to discover differences between classes of different releases.

The Semantic Measures Library (SML) is an open source Java library developed for semantic measures computation and analysis. The SML library and the associated toolkit can be used to compute semantic similarity and semantic relatedness between concepts or other entities that are semantically annotated using concepts defined in an ontology.

SEMILAR (Rus *et al.*, 2013) is an open source java library that comes with various similarity methods based on WordNet, Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA), etc. In addition, the similarity methods work in different granularities: word-to-word, sentence-to-sentence, or text-to-text.

WordNet::Similarity (Pedersen *et al.*, 2004), and UMLS::Similarity (McInnes *et al.*, 2009) are both open source Perl modules in which a variety of semantic similarity and relatedness measures are implemented for assessing semantic similarity between concepts of WordNet (Miller, 1995) and UMLS (2013) respectively.

In fact, the latest version of the library SimPack (v0.91) was released in 2008 with no recent update of maintenance. Both SML and SEMILAR were still in initial development stage during the experimental study of this thesis. On the contrary, both WordNet::Similarity UMLS::Similarity demonstrated stability, reliability, and effectiveness through many applications (Séaghdha, 2009; McInnes *et al.*, 2011). For reasons related to the application domain, we will use the module UMLS::Similarity in the semantic similarity engine to assess semantic similarity between concepts in UMLS.

In order to reduce the side effects of using this module in our system, we modify the system presented in Figure 30 and add a database to the system (see Figure 42). This database works as a cache to store calculated semantic similarities. This database contains the identifiers of the concepts and the assessed similarity between them using a particular similarity measure.

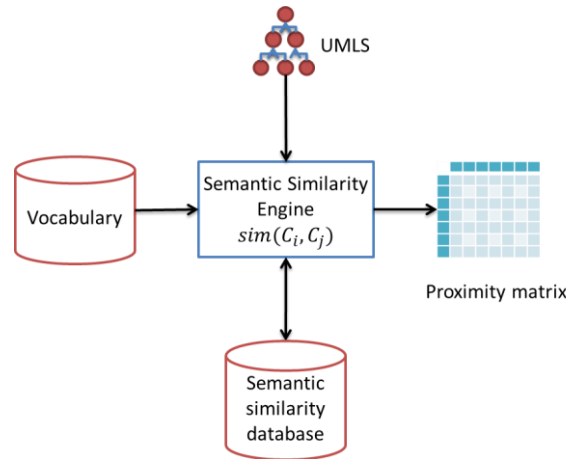


Figure 42. Semantic similarity engine with a cache database for building proximity matrix

Figure 43 illustrates the activity diagram of the semantic similarity engine: first, it chooses a pair of concepts between which to assess the semantic similarity. Then, it queries the database searching for an entry that corresponds to the compared concepts and the used measure with other configurations. If an entry exists, this means that the system has calculated the similarity before using the same configurations, so it assigns the value retrieved in the entry to the corresponding case in the proximity matrix. If no entry corresponds to the concepts, the system calculates the similarity between them in UMLS using the module UMLS::Similarity that we will present in next section.

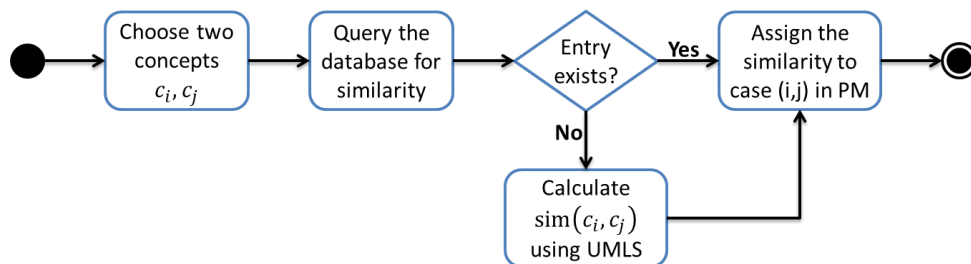


Figure 43. Activity diagram of the semantic similarity engine

6.2.2 UMLS::Similarity

UMLS::Similarity is a Perl module that provides an API and a command line program to estimate the semantic similarity between concepts using their *Concept Unique Identifiers* (CUIs) in UMLS. As for version UMLS-Similarity-1.33 that we used in this work, UMLS-Similarity contains nine semantic similarity measures:

- Measures based on Path:
 - path: is the reciprocal of the number of nodes between two concepts. It returns values that range between zero and one.
 - **cdist** (Caviedes *et al.*, 2004): is an adapted version of the measure proposed in (Rada *et al.*, 1989) to UMLS. It counts the number of edges between the compared concepts. Its range is between zero and twice the depth of the ontology.

- Measures based on Path and Depth
 - **wup** (Wu *et al.*, 1994): is twice the depth of the concepts' msca divided by the sum of the depths of the concepts. Its range is between zero and one.
 - **lch** (Leacock *et al.*, 1998): is the negative log of the shortest path between two concepts divided by twice the total depth of the ontology. Its range is unbounded and depends on the depth of the ontology.
 - **zhong** (Zhong *et al.*, 2002): is the sum of the difference between the milestone of the msca and each of the concepts. The milestone is a calculated factor and is related to the specificity of concepts. Its range is between zero and one.
 - **nam** (Al-Mubaid *et al.*, 2006): is the log of a formula of the shortest distance between the two concepts and the depth of the ontology minus the depth of the concepts msca. Its range depends on the depth of the ontology.
- Measures based on IC
 - **res** (Resnik, 1995): is the IC of the msca of the concepts. Its range is between zero and log the size of the ontology.
 - **jcn** (Jiang *et al.*, 1997): is twice the similarity of Resnik divided by the sum of the ICs of the compared concepts. Its range is between zero and one.
 - **lin** (Lin, 1998): is the sum of the IC of the concepts minus twice the similarity of Resnik. Its range is between zero and twice log the size of the ontology.

In order to deploy the module UMLS::Similarity in the semantic similarity engine, it is necessary to install two other components (see Figure 44): a local installation of the ontology UMLS® in a MySQL database and UMLS::Interface. UMLS::Interface is a Perl module that provides an API to access and explore UMLS®. Some of its programs return information about a concept using its CUI whereas others return information about paths like the paths between a concept and the root.

In the same way as we explained earlier and for reasons related to efficiency, we will limit our experimentations to five state of the art ontology-based semantic similarity measures: **cdist**, **wup**, **lch**, **zhong**, **nam**. Furthermore, we choose to limit the access of our system on UMLS® to SNOMED-CT® that is one of the largest and the broadest semantic resources integrated in UMLS®. This implies that the compared concepts belong certainly to SNOMED-CT®.

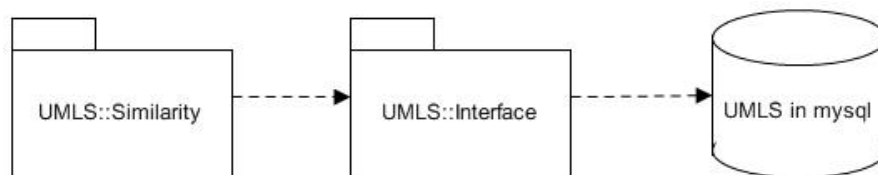


Figure 44. Components inside the semantic similarity engine for the medical domain

6.3 Conclusion

This section presented and compared some tools developed in the medical domain for text to concept mapping and for semantic similarity, all applied on UMLS or parts of it.

The first part of this section introduced four different tools for mapping text to concepts in the UMLS or to the resources it unifies. Table 25 presents a comparison between these tools through their principles, advantages and disadvantages.

Tool	Basic principle	Advantages	Disadvantages
PubMed ATM ("PubMed Tutorial," 2013)	Exact dictionary lookup	-Simplicity -High precision	-Low recall -Looks for exact match to every word
MaxMatcher (X. Zhou <i>et al.</i> , 2006)	Approximate dictionary lookup	-Better recall -Good precision	-Looks for exact match to significant words
MGREP (Dai, 2008)	Matches text to all concepts variations using a radix-tree	-High precision -Good recall -Linguistic analysis of concepts	-Less recall -Precision depends on treated data
MetaMap (Aronson <i>et al.</i> , 2010)	Applies Rigorous linguistic analysis of text	-Good precision -High recall -Uses linguistic of text	-Time-consuming linguistic analyses

Table 25. Comparing four tools for text to UMLS concept mapping

Exact dictionary lookup base tool, PubMed ATM, uses a very simple technique for matching the exact terms as found in text that reduces the recall of mappings. MaxMatcher, the second tool presented in this section, chooses to match text to the most important terms in concepts trying to overcome the limitation of the previous tool through an *approximate dictionary lookup*. MGREP and MetaMap deploy sophisticated linguistic analysis that improves mapping effectiveness. MetaMap, applies this analysis on text which slows down its process. On the other hand MGREP chooses to apply analysis on concepts a priori and keeps mapping-time algorithms less sophisticated. Consequently, MGREP seems to be more efficient than MetaMap in mapping. According to authors in (Shah *et al.*, 2009; Aronson *et al.*, 2010), MGREP is more precise than MetaMap that demonstrated a higher recall. This evaluation may change according to the used dataset in evaluation.

Finally, we choose to use MetaMap for the text to concept mapping in this work accepting its weakness in real-time processing as compared to its effectiveness in recognizing UMLS concepts in medical text.

The second part of this section detailed the semantic similarity engine that builds proximity matrices for the medical domain in which we intend to realize our experiments. We choose UMLS[®] as a semantic resource and UMLS::Similarity module to assess the semantic similarity between its concepts. UMLS::Interface provides an API with useful utility programs as an intermediate between UMLS[®] and UMLS::Similarity. We selected five ontology-based measures that we intend to use in further experiments.

We discussed and argued some technical choices that we made in order to implement the previously presented scenarios in the medical domain. These choices will be applied in experiments in next chapter.

7 Conclusion

In general, text classification is tackled using syntactic and statistical information only. Moreover, the conventional BOW ignores semantics that reside in text and suffers from ambiguities, redundancy and orthogonality in treating features. In this chapter, we proposed generic frameworks and approaches for involving semantics in different steps of supervised text classification: indexing through *conceptualization*, *enriching text representation* and in *assessing similarities between text documents*. In this aim, this chapter presented a conceptual framework for involving semantics in text classification. We discussed using concepts in conceptualization and semantic similarities between concepts in the other approaches. All of the four approaches can be applied through four scenarios. In addition, this chapter presented many tools for the medical domain that we found effective in realizing text conceptualization and in assessing semantic similarities between concepts as well.

We have already compared the three techniques: NB, SVM and Rocchio in chapter 2. Here we add a criteria describing to what extent the classifier accepts semantics integration in the process of classification. In fact, Rocchio is the only classifier, among those evaluated in this work that can deploy semantics in class prediction through new semantic similarity measures. These measures depend on concepts to which text documents are mapped and the relations among them in the semantic resource. Moreover, Rocchio is the only one with a vector like classification model that accepts semantic enrichment. On the other hand, NB and SVM accept the integration of semantics in text representation through conceptualization.

Next chapter investigates the influence of semantics on classifiers effectiveness especially for difficult cases like large classes and poorly populated classes through experimental study on Ohsumed corpus. We will deploy and validate the approaches proposed for involving semantics in indexing using NB, SVM and Rocchio. Enriching text representation and semantic text-to-text similarity measures for class prediction are tested using Rocchio technique only.

CHAPTER 5: SEMANTIC TEXT CLASSIFICATION: EXPERIMENT IN THE MEDICAL DOMAIN

Table of contents

1	Introduction.....	141
2	Experiments applying scenario1 on Ohsumed using Rocchio, SVM and NB.....	142
2.1	Platform for supervised classification of conceptualized text.....	142
2.1.1	Text Conceptualization task	143
2.1.2	Indexing task.....	144
2.1.3	Training and classification tasks	147
2.2	Evaluating Results	147
2.2.1	Results using Rocchio with Cosine	148
2.2.2	Results using Rocchio with Jaccard	150
2.2.3	Results using Rocchio with KullbackLeibler	152
2.2.4	Results using Rocchio with Levenshtein	154
2.2.5	Results using Rocchio with Pearson	156
2.2.6	Results using NB.....	158
2.2.7	Results using SVM	160
2.2.8	Comparing MacroAveraged F1-Measure of the Classification Techniques	162
2.2.9	Comparing F1-Measure of the Classification Techniques for each class.....	164
2.2.10	Conclusion	168
3	Experiments applying scenario2 on Ohsumed using Rocchio.....	169
3.1	Platform for supervised text classification deploying Semantic Kernels.....	169
3.1.1	Text Conceptualization task	170
3.1.2	Proximity matrix	170
3.1.3	Enriching vectors using Semantic Kernels	172
3.2	Evaluating results.....	172
3.2.1	Observations.....	173
3.2.2	Analysis and conclusion.....	174
4	Experiments applying scenario3 on Ohsumed using Rocchio.....	176
4.1	Platform for supervised text classification deploying Enriching Vectors	176
4.1.1	Enriching Vectors	177
4.2	Evaluating results.....	177
4.2.1	Results using Rocchio with Cosine	177
4.2.2	Results using Rocchio with Jaccard	179
4.2.3	Results using Rocchio with Kulback	180
4.2.4	Results using Rocchio with Levenshtein.....	181
4.2.5	Results using Rocchio with Pearson	181
4.2.6	Conclusion	183
5	Experiments applying scenario4 on Ohsumed using Rocchio.....	185
5.1	Platform for supervised text classification deploying Semantic Text-To-Text Similarity Measures	185
5.1.1	Semantic Text-To-Text Similarity Measures	185
5.2	Evaluating results.....	186
5.2.1	Results using AvgMaxAssymldf.....	186
5.2.2	Results using AvgMaxAssymTFIDF	187
5.2.3	Conclusion	188
6	Conclusion	190

1 Introduction

In previous chapter, we presented a framework for integrating semantics in the process supervised text classification. We suggested four different scenarios to apply this framework involving semantics before and after indexing and during prediction. Previous chapter introduced useful tools for its implementation in the medical domain including tools for text to concept mapping and others for assessing semantic similarity between concepts in the semantic resource.

In this chapter we present an experimental study to investigate the influence of semantics on classifier effectiveness especially for difficult cases identified previously in this work (see chapter 2) like large classes and poorly populated classes. We intend to integrate semantics before indexing through *Conceptualization* and after indexing through enrichment using either *Semantic Kernels* or *Enriching Vectors*. Moreover, we'll investigate the influence of *Semantic Text-To-Text Similarity* on text classification. The corpus we'll use in all experiments is Ohsumed, the well-known corpus in the medical domain. We choose to use UMLS as the semantic resource for the implemented platforms.

Each section presents first the platform with some technical details and then presents and analyzes the detailed results in order to give some recommendations on the use of semantics in supervised text classification and particularly in the medical domain. In second section, we test Rocchio (5 variants), SVM and NB using conceptualization whereas in the rest of this chapter we apply semantic approaches on Rocchio due to its extendibility and its vector like classification model as justified in previous chapter. The architecture of our platforms is modular and generic; its components can be modified and even replaced.

This chapter is organized as the following: section 2 presents experiments on Ohsumed after conceptualization in a platform implementing the first scenario seen in previous chapter and using three different classification techniques. Section 3 presents experiments on Ohsumed using Semantic Kernels for enrichment and Rocchio for classification; this section applies the second scenario seen in previous chapter. Section 4 presents experiments on Ohsumed using Enriching Vectors for enrichment and Rocchio for classification and implementing the third scenario of the previous chapter. Section 5 presents experiments on Ohsumed using semantic similarity measures for class prediction with Rocchio implementing the fourth scenario seen in the previous chapter.

2 Experiments applying scenario1 on Ohsumed using Rocchio, SVM and NB

In these experiments we intend to assess the impact of conceptualization on text classification applied in the medical domain. We use different classification techniques and try to compare this impact on each of them using different conceptualization strategies. These experiments apply the first scenario of the previous chapter.

This section presents the platform for our experiments in some details and then presents the results from different points of views and concludes with some recommendation on the use of conceptualization in the context of text classification.

2.1 Platform for supervised classification of conceptualized text

This section presents an experimental platform for assessing the impact of different conceptualization strategies on text classification, using three classical text classification techniques: Rocchio (5 variants), SVM and NB. This platform is illustrated in Figure 45. The upper part of the figure concerns training phase in which a classification technique learns a classification model on the index of the conceptualized corpus. Whereas, the lower part illustrates the classification phase in which the same classification technique uses the classification model in order to predicate the class of each test document. This document is represented using the same vocabulary and weighting scheme as those used to represent the training corpus.

The architecture of our platform is modular and generic; its components can be modified and even replaced. In this work we use three different classical classification techniques, Rocchio, NB and SVM, to realize Training and Classification phases.

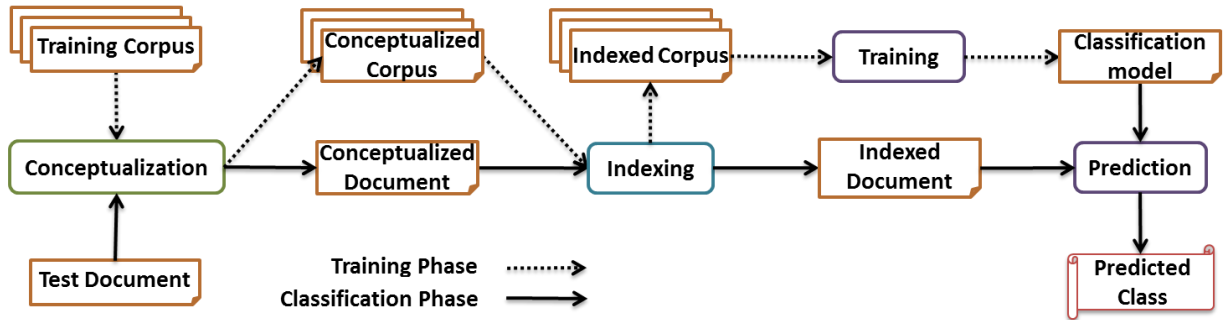


Figure 45. The architecture of a platform for conceptualized text classification.

This section presents first, the text conceptualization task performed on the Ohsumed corpus using UMLS[®] (2013) and MetaMap tool (Aronson *et al.*, 2010), according to different strategies. Then, it presents some details on the indexing, training and classification tasks as well. Then, it presents classification results obtained with each of the three classical classification methods: Rocchio (5 variants), SVM and NB for each of these conceptualization strategies. Finally, this section analyzes and discusses the obtained results.

2.1.1 Text Conceptualization task

During the conceptualization task, different strategies can be implemented as previously described (adding concepts, partial conceptualization and complete conceptualization). Furthermore, according to MetaMap text-to-concept matching results, we can choose two complementary strategies:

- Best concept strategy. Choosing the best concept among several candidate concepts that are matched to the text. This depends on a matching score computed by MetaMap (Aronson *et al.*, 2010).
- All concepts strategy. All candidate concepts are kept.

Candidates resulting from matching have many properties like name, unique identifier, semantic type, definition and so on. In this work we choose to use the concept name or the concept Id. In fact, during the tokenization step, the concept Id is considered as a single token so it stays intact. Concept names, being sometimes compound words, can be broken down during tokenization when applied on a text that was conceptualized using concept name strategy. In this work, conceptualization is done using all combinations of the different strategies (12 combinations).

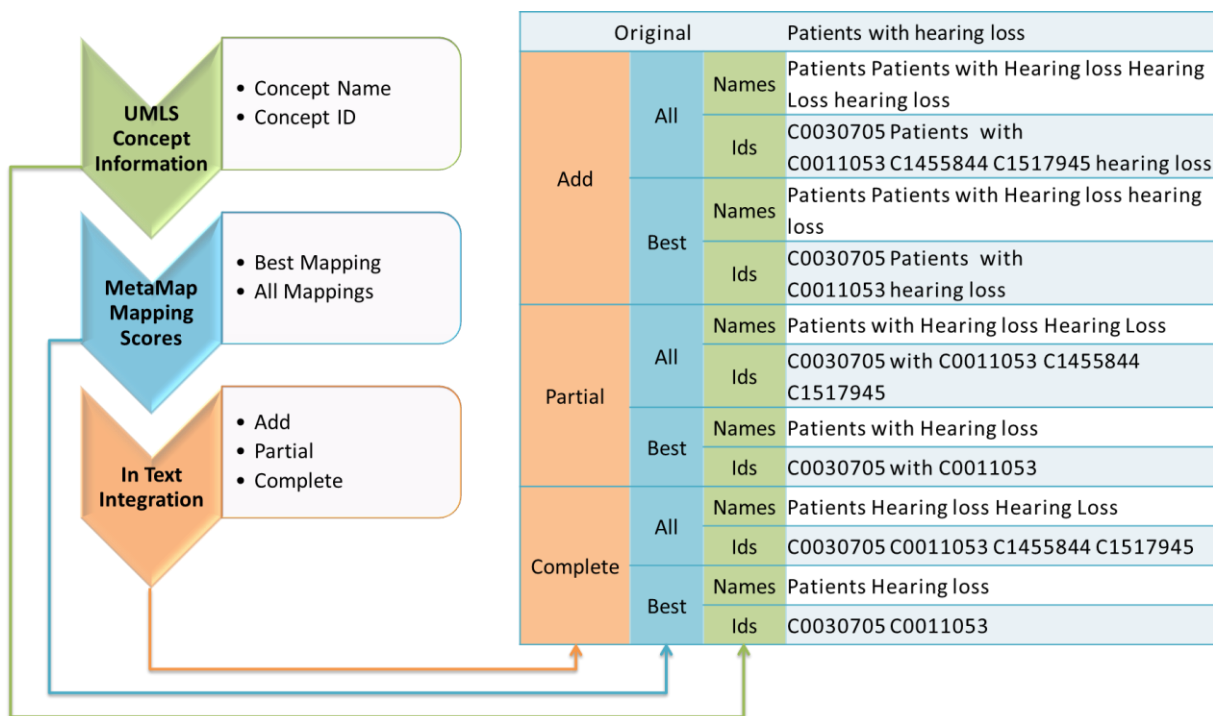


Figure 46. 12 strategies for text conceptualization using MetaMap: a walk through an example. For the utterance "with hearing loss" we chose to use a maximum of two mappings to avoid confusion.

Figure 46 illustrates the twelve different conceptualization strategies resulting from matching:

- Two types of information for each UMLS concept: Name or Identifier.
- Two strategies to choose the concepts from the mapping list returned by MetaMap; choosing either the best or all the mappings.

- Three strategies for integrating semantics into text adding, substituting (partial) or keeping only concepts (complete).

The same textual example is used in a previous illustration (MetaMapFigure) where mapping results are detailed.

To process a text document, MetaMap treat text using rigorous linguistic analysis and then queries UMLS for Mappings. Finally, these mappings are integrated into the original text according to a particular conceptualization strategy. These steps are illustrated in Figure 47.

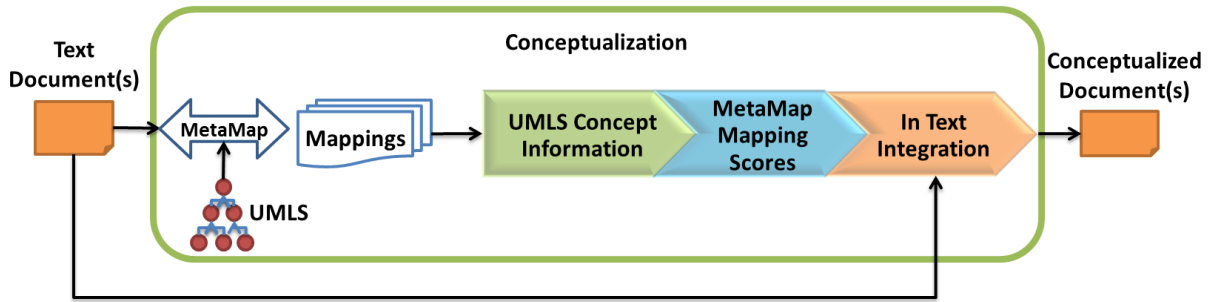


Figure 47. Conceptualization: the process step by step

2.1.2 Indexing task

In general, Indexing processes text and constitutes a vector of features that represents text contents. Our system indexes text through the following steps (see Figure 48): First, it transforms text to a vector of words considering word frequency in text as its weight. For this step, the system uses Lucene with Rocchio as classification technique or Weka (Hall *et al.*, 2009) with SVM and NB. Then, it eliminates stop words from these vectors and applies stemming to words using Porter Stemmer (Porter, 1980). The system applies finally the weighting scheme known by TFIDF (Term Frequency/Inverse Document Frequency) and keeps the first 2000 terms by class of documents. The vocabulary of terms collected on the training corpus constitutes the feature space into which the indexer projects every new document presented to the system.

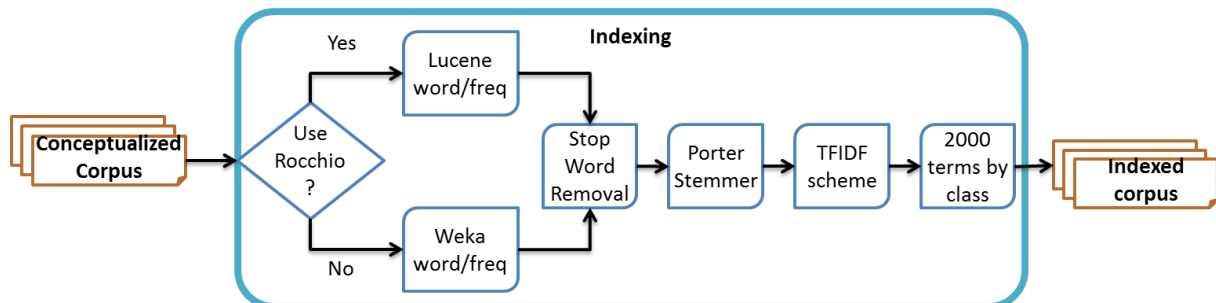


Figure 48. Indexing process: step by step

The proposed platform in Figure 45, applies indexing on conceptualized text. The result of conceptualizing text differs according to the chosen strategy. For example, the last two vectors in the Table 26 correspond to the strategies (Complete+Best+Names) and (Complete+Best+Ids)

respectively. In both strategies, only two concepts are integrated in the final text: “Patients” with Id:C0030705 and “Hearing loss” with Id:C0011053. After indexing, the concept “Patients” is indexed in the same way in both cases whereas the concept “Hearing loss” is indexed using two words in the first case and its unique Id in the second case. This is due to the fact that conventional indexing doesn’t take into consideration composed words. This is one of the drawbacks of the classical BOW that we try to overcome using IDs in conceptualization in order to impose indexing composed words altogether.

Patients with hearing loss	patients	with	hearing	loss	C0030705	C0011053	C1455844	C1517945
	1	1	1	1	0	0	0	0
Patients Patients with Hearing loss Hearing Loss hearing loss	patients	with	hearing	loss	C0030705	C0011053	C1455844	C1517945
	2	1	3	3	0	0	0	0
C0030705 Patients with C0011053 C1455844 C1517945 hearing loss	patients	with	hearing	loss	C0030705	C0011053	C1455844	C1517945
	1	1	1	1	1	1	1	1
Patients Patients with Hearing loss hearing loss	patients	with	hearing	loss	C0030705	C0011053	C1455844	C1517945
	2	1	2	2	0	0	0	0
C0030705 Patients with C0011053 hearing loss	patients	with	hearing	loss	C0030705	C0011053	C1455844	C1517945
	1	1	1	1	1	1	0	0
Patients with Hearing loss Hearing Loss	patients	with	hearing	loss	C0030705	C0011053	C1455844	C1517945
	1	1	2	2	0	0	0	0
C0030705 with C0011053 C1455844 C1517945	patients	with	hearing	loss	C0030705	C0011053	C1455844	C1517945
	0	1	0	0	1	1	1	1
Patients with Hearing loss	patients	with	hearing	loss	C0030705	C0011053	C1455844	C1517945
	1	1	1	1	0	0	0	0
C0030705 with C0011053	patients	with	hearing	loss	C0030705	C0011053	C1455844	C1517945
	0	1	0	0	1	1	0	0
Patients Hearing loss Hearing Loss	patients	with	hearing	loss	C0030705	C0011053	C1455844	C1517945
	1	0	2	2	0	0	0	0
C0030705 C0011053 C1455844 C1517945	patients	with	hearing	loss	C0030705	C0011053	C1455844	C1517945
	0	0	0	0	1	1	1	1
Patients Hearing loss	patients	with	hearing	loss	C0030705	C0011053	C1455844	C1517945
	1	0	1	1	0	0	0	0
C0030705 C0011053	patients	with	hearing	loss	C0030705	C0011053	C1455844	C1517945
	0	0	0	0	1	1	0	0

Table 26. Transform the phrase “Patients with hearing loss” into word/frequency vector before and after conceptualization using the 12 conceptualization strategies.

Most classification techniques are sensitive to the number of features since it affects their efficiency or their effectiveness. Thus, we carried out experiments on the Ohsumed corpus using Rocchio with Cosine. Two versions of Ohsumed were concerned in these experiments: the original text and the conceptualization version according to the strategy (Complete+Best+Ids). The goal of these experiments is to assess the effect of the number of features on classification effectiveness. So for each class, we limit the vocabulary after training phase to n features with maximum TF/IDF values in the corpus (Özgür *et al.*, 2005). We varied this n from 100 to 4000 and then tested the classifier on each resulting model and recorded the F1-measure for each n .

Figure 49 illustrates the effect of varying n on the textual corpus. In fact, the value of F1-measure increases with the increase of n which means that the more features the classifier had the easier it identifies the classes. However, the increase in F1-measure become marginal when $(500 < n < 2000)$ which means that the rest of Rocchio’s features aren’t vital to

classification. Notice that the classifier shows a relatively constant performance when ($n > 2000$).

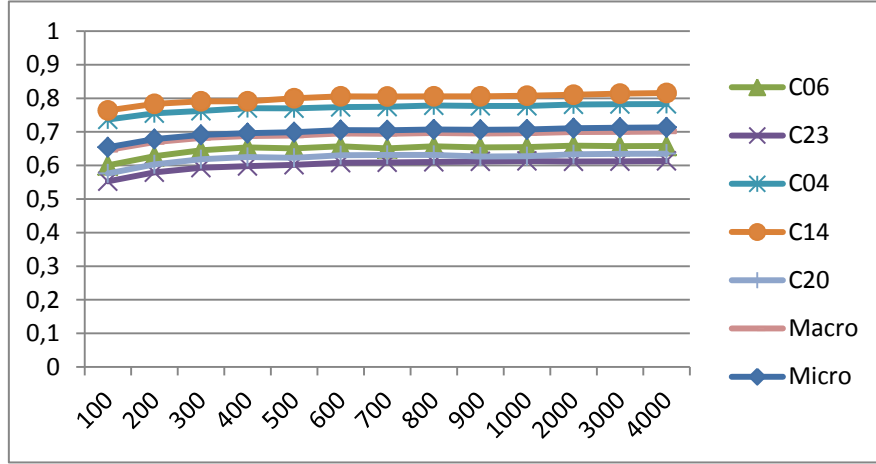


Figure 49. Evaluating the effect of vocabulary size that varies from [100 to 4000] features on classification results (F1-measure) using Rocchio with Cosine on Ohsumed textual corpus

Figure 50 illustrates the effect of varying n on the conceptualized corpus. The difference between these experiments and the previous ones is that features are concept Ids and not normalized words. In fact, the value of F1-measure increases with the increase of n which means that the more features the classifier had the easier it identifies the classes. However, the increase in F1-measure become marginal when ($800 < n < 2000$) which means that the rest of Rocchio's features are not vital to classification. Notice that the classifier shows a relatively constant performance when ($n > 2000$).

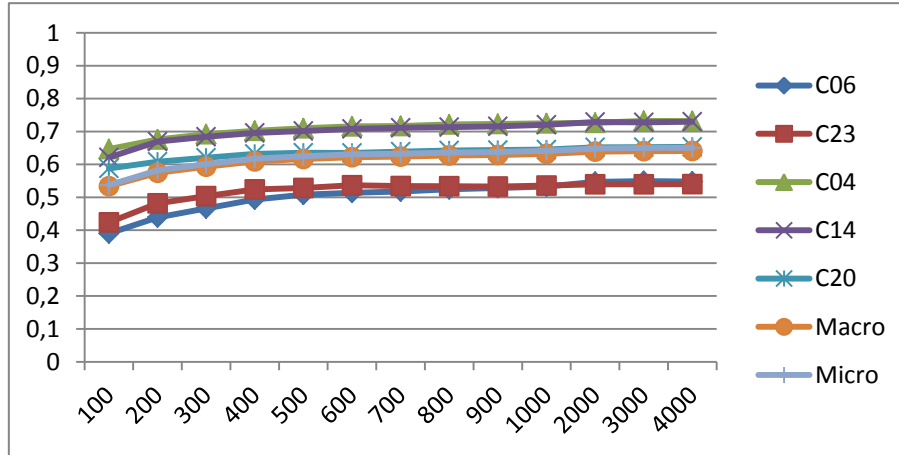


Figure 50. Evaluating the effect of vocabulary size that varies from [100 to 4000] features on classification results (F1-measure) using Rocchio with Cosine on Ohsumed conceptualized corpus according to the strategy ("Complete", "Best", "Ids").

As a conclusion, using Ids as features makes the vectors sparser in the feature space and makes the classifier need more features in order to identify and to distinguish the different classes. In the rest of this work we choose the value ($n = 2000$) as a compromise between efficiency and effectiveness; we will limit the vocabulary size to (2000) term by class in the forthcoming experiments.

2.1.3 Training and classification tasks

In our experiments, we tested seven techniques on the Ohsumed corpus before and after conceptualization. In each test, the system uses one of the twelve conceptualization strategies presented earlier in section 2.1.2.

During Training phase the training corpus is prepared for training the classifier resulting in a classification model whereas during the Classification phase the test corpus is prepared in order to attribute classes to its document by the classifier using the learned classification model.

We used three classical classification methods in our experiments:

- Rocchio, using five different similarity measures: Cosine, Jaccard, KullbackLeibler, Levenshtein and Pearson (A. Huang, 2008). This creates five different variants of Rocchio that use the same classification model and differ in predicting criterion at the classification phase only.
- SVM, using the library LIBSVM (Chang *et al.*, 2011) wrapped in WLSVM (EL-Manzalawy *et al.*, 2005) and integrated in Weka (Hall *et al.*, 2009).
- NB, using the platform Weka (Hall *et al.*, 2009).

In all experiments, methods are evaluated through Holdout Validation. F1-Measure (Sokolova *et al.*, 2009) is the used criterion for performance evaluation and comparison.

2.2 Evaluating Results

This section presents in details the results of experiments using the previous platform. These experiments count $(7 * (12 + 1)) = 91$ tests for seven classifiers on the original textual Ohsumed and the conceptualized corpus by means of MetaMap according to twelve different conceptualization strategies as well.

Next seven subsections present the observations and the analysis of results using each of the seven classifiers (5 variations of Rocchio, SVM and NB) on the five classes of documents (C04, C06, C14, C20, C23). The two last columns of each result table present Micro and Macro averaged F1-measure obtained for each pair of classification techniques and conceptualization strategies. In Micro-averaging, F1-measure is computed globally over all category documents, whereas in Macro-averaging it is equal to the average of locally calculated F1-measures for each class. We evaluate the significance of differences between classifiers' performance on text and on conceptualized text according to McNemar statistical test (Kuncheva, 2004) with a level of significance ($\alpha = 0.05$).

The eighth subsection compares the seven classifiers among them on Ohsumed before and after conceptualization using MacroAverage F1-measure. We evaluate the significance of differences between classifiers' performance on text and on conceptualized text according to t-test (Yang *et al.*, 1999) statistical test with a level of significance ($\alpha = 0.05$).

The ninth subsection compares the seven classifiers among them on Ohsumed before and after conceptualization on different classes of documents. The goal is to identify the classes where maximum improvements occurred.

The section concludes with discussion pointing to some recommendations on best practice in enriching text with semantics for effective text classification.

2.2.1 Results using Rocchio with Cosine

2.2.1.1 Observations

According to results illustrated in Table 27, the F1-measure obtained from applying Rocchio with Cosine similarity measure on the original Ohsumed corpus varied from (61.17%) to (81.04%) for classes (C23, C14) respectively. We report improvements in classification with four conceptualization strategies: (Add+Best+Names), (Add+Best+Ids), (Partial+Best+Names), (Complete+Best+Names). These improvements increased the Macro F1-measure using the first two strategies only: (Add+Best+Names) and (Add+Best+Ids). Note that the most effective conceptualization strategy is (Add+Best+Names) using Rocchio with Cosine as a similarity measure whereas the improvement reported for (Add+Best+Ids) is minor. Deploying both strategies (Partial+Best+Names) and (Complete+Best+Names) improved classifying particular classes only, this limited the deterioration of the overall performance.

Corpus \ Class			C04		C06		C14		C20		C23		Macro		Micro	
Original			78.16		65.92		81.04		63.35		61.17		69.93		71.09	
Add	All	Names	72.06	-7.80*	62.05	-5.87	76.22	-5.95*	57.00	-10.02*	57.06	-6.71	64.88	-7.22*	66.01	-7.14
		Ids	74.05	-5.25*	62.75	-4.81	77.64	-4.20*	61.99	-2.14*	59.77	-2.28*	67.24	-3.84*	68.18	-4.09
	Best	Names	78.19	+0.04	67.99	+3.13*	81.17	+0.16*	64.04	+1.09	62.05	+1.44*	70.69	+1.08	71.63	+0.76
		Ids	77.58	-0.74	65.27	-0.99	80.63	-0.51*	64.68	+2.11	61.61	+0.72*	69.95	+0.04	71.01	-0.11
Partial	All	Names	70.35	-9.99*	59.27	-10.09*	74.87	-7.62*	53.90	-14.91*	55.84	-8.72	62.85	-10.13*	64.23	-9.65
		Ids	73.28	-6.23*	61.44	-6.80*	77.21	-4.74*	61.59	-2.77*	59.29	-3.07*	66.56	-4.81*	67.54	-5.00
	Best	Names	76.78	-1.77	66.93	+1.54*	79.66	-1.71*	60.97	-3.76*	61.34	+0.29*	69.14	-1.13	70.18	-1.27
		Ids	71.27	-8.81*	53.97	-18.14*	70.51	-13.00*	62.54	-1.27*	53.30	-12.86*	62.32	-10.88*	63.18	-11.12
Complete	All	Names	70.37	-9.96*	59.23	-10.15*	74.79	-7.72*	53.95	-14.84*	55.72	-8.90	62.81	-10.18*	64.19	-9.71
		Ids	73.34	-6.17*	61.58	-6.58*	77.31	-4.61*	61.75	-2.51*	59.36	-2.96*	66.67	-4.66*	67.64	-4.85
	Best	Names	76.87	-1.65	67.41	+2.26*	79.52	-1.88*	61.28	-3.26*	60.75	-0.68*	69.17	-1.09	70.12	-1.35
		Ids	71.82	-8.10*	54.74	-16.96*	70.41	-13.12*	61.58	-2.79*	53.73	-12.16*	62.46	-10.68*	63.42	-10.78

Table 27. Results of applying Rocchio with Cosine similarity measure to Ohsumed corpus and to the results of its conceptualization according to 12 conceptualization strategies. (*) denotes significance according to McNemar test. Values in the table are percentages.

The strategy (Add+Best+Names) improved the performance of Rocchio with Cosine similarity measure by a percentage that varies from (0.04%) for the class (C04) to (3.13%) for the class (C06). The absolute value of F1-measure varied from (62.05%) to (81.17%) for classes (C23, C14) respectively. The second strategy (Add+Best+Ids) increased the F1-measure by (2.11%) and (0.72%) which resulted in the values (64.68%, 61.61%) for (C20) and (C23) respectively. The strategy (Partial+Best+Names) increased the F1-measure of both classes (C06,

C23) by (1.54%, 0.29%) resulting in (66.93%, 61.34%) respectively. Finally, the strategy (Complete+Best+Names) increased the F1-measure of (C06) by (2.26%) resulting in (67.41%).

2.2.1.2 Analysis

From previous observations we conclude that maximum increase in F1-Measure (3.13%) was obtained for the class (C06) using the strategy (Add+Best+Names). In fact, this class is one of the least populated classes and on which we obtained a relatively low F1-Measure (65.92%) using the original corpus. This means that Rocchio using Cosine did not learn an effective classification model on the original text using the relatively small set of training documents of this class. In addition, Cosine may not detect enough common features between the classification model and the new documents related to C06. Thus, text *Conceptualization* enhanced learning and prediction capabilities of Rocchio with Cosine.

Previous reported improvements at class level influenced the MacroAveraged F1-Measure with a gain of (0.04%) and (1.08%) using strategies (Add+Best+Ids) and (Add+Best+Names) respectively. Note that we have no evidence that the overall performance of Rocchio using Cosine on the original corpus is significantly different from its performance on the corpus after applying either strategy according to McNemar test.

In fact, enriching text by adding the names of the best mapped concepts is useful to classifying different classes of documents using Rocchio with Cosine. However, enriching text by adding the Ids of the best mappings is less interesting considering the overall performance yet relatively effective for classes (C20, C23). On the other hand, using names of best mappings with Complete or Partial demonstrate improvements at class level for (C06, C23) and (C06) respectively. In fact, Rocchio with Cosine seems to be highly dependent on text statistics and thus replacing text with Ids of corresponding concepts disturbs learning and classification and results in deterioration in its effectiveness.

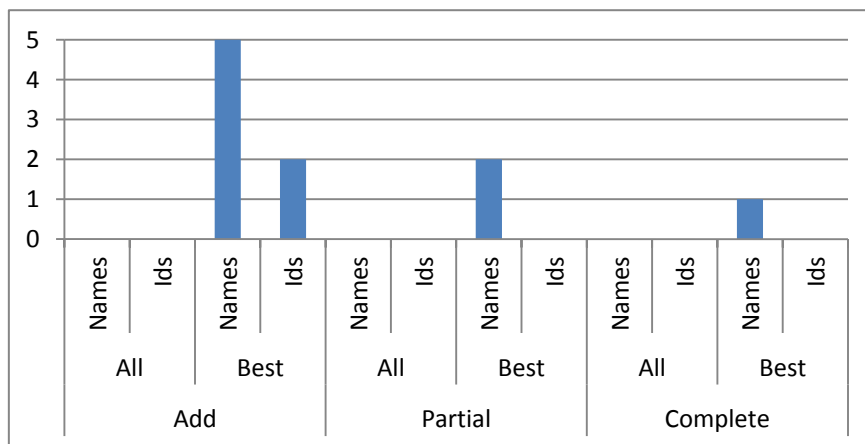


Figure 51. Number of classes with improved F1-Measure on conceptualized text compared with the original text using Rocchio with Cosine similarity measure

According to Figure 51, using the strategy (Add+Best+Names) increased F1-Measure of the five classes which improved the overall performance of Rocchio with Cosine. This improvement is the maximum increase obtained among all other strategies that results in a MacroAveraged F1-measure of (70.69%) as presented formerly in Table 27. Note that for each of the three

strategies (Add, Partial, Complete) maximum number of improved classes after conceptualization is obtained using the names of the best mapped concepts. This is due to Rocchio's dependency on text statistics that is fortified using names rather than using Ids.

2.2.2 Results using Rocchio with Jaccard

2.2.2.1 Observations

According to results illustrated in Table 28, the F1-measure obtained from applying Rocchio with Jaccard similarity measure on the original Ohsumed corpus varied from (56.68%) to (82.02%) for classes (C23, C14) respectively. Nine conceptualization strategies helped Rocchio improve its performance: (Add+All+Ids), (Add+Best+Names), (Add+Best+Ids), (Partial+All+Ids), (Partial+Best+Names), (Partial+Best+Ids), (Complete+All+Ids), (Complete+Best+Names), (Complete+Best+Ids). These improvements appeared at the MacroAveraged level using the Names of the Best mapping with any of the three strategies for text integration. Among these three strategies, (Add+Best+Names) is the most effective as it improves the classification of the five classes which is not the case for the other two strategies.

The strategy (Add+All+Ids) increased the F1-measure by (0.30%) which resulted in the value (56.85%) for (C23). The strategy (Add+Best+Names) improved the performance of Rocchio with Jaccard similarity measure by a percentage that varies from (0.40%) for the class (C14) to (3.15%) for the class (C23). The absolute value of F1-measure varied from (58.47%) to (82.35%) for classes (C23, C14) respectively. The strategy (Add+Best+Ids) increased the F1-measure by (2.92%) which resulted in the value (64.81%) for (C20).

Corpus\ Class			C04		C06		C14		C20		C23		Macro		Micro	
Original			78.23		61.74		82.02		62.98		56.68		68.33		70.12	
Add	All	Names	72.43	-7.42*	61.65	-0.15*	77.51	-5.50*	57.32	-8.98*	55.90	-1.39*	64.96	-4.93*	66.39	-5.32
		Ids	74.95	-4.19*	59.15	-4.21*	79.18	-3.47*	62.46	-0.82*	56.85	+0.30*	66.52	-2.65*	68.02	-3.00
	Best	Names	78.83	+0.76	64.45	+4.38	82.35	+0.40	63.69	+1.13	58.47	+3.15*	69.56	+1.79*	71.21	+1.55
		Ids	77.96	-0.35	59.28	-3.98	81.67	-0.43*	64.81	+2.92	54.66	-3.57*	67.68	-0.96	69.62	-0.72
Partial	All	Names	71.19	-9.00*	58.71	-4.92*	75.86	-7.51*	55.56	-11.78*	54.84	-3.25*	63.23	-7.46*	64.71	-7.73
		Ids	74.09	-5.29*	59.21	-4.10*	78.71	-4.04*	61.98	-1.58*	57.61	+1.63*	66.32	-2.94*	67.70	-3.46
	Best	Names	77.34	-1.14	65.01	+5.29*	80.73	-1.57*	61.93	-1.66*	58.47	+3.15*	68.70	+0.53	70.16	+0.06
		Ids	71.34	-8.81*	51.13	-17.19*	71.52	-12.81*	63.98	+1.59*	45.36	-19.97*	60.67	-11.22*	62.10	-11.44
Complete	All	Names	71.09	-9.13*	58.56	-5.15*	75.82	-7.56*	55.65	-11.64*	54.91	-3.13*	63.21	-7.50*	64.67	-7.78
		Ids	73.99	-5.41*	59.28	-3.99*	78.88	-3.84*	61.92	-1.67*	57.78	+1.93*	66.37	-2.87*	67.74	-3.40
	Best	Names	77.26	-1.24	65.67	+6.37*	80.54	-1.81*	61.57	-2.23*	59.11	+4.28*	68.83	+0.73	70.26	+0.20
		Ids	71.39	-8.74*	52.18	-15.48*	70.99	-13.45*	64.19	+1.93*	46.67	-17.66*	61.09	-10.60*	62.40	-11.02

Table 28. Results of applying Rocchio with Jaccard similarity measure to Ohsumed corpus and to the results of its conceptualization according to 12 conceptualization strategies. (*) denotes significance according to McNemar test. Values in the table are percentages.

The strategy (Partial+All+Ids) increased the F1-measure by (1.63%) which resulted in the value (57.61%) for (C23). The strategy (Partial+Best+Names) increased the F1-measure by (5.29%) for the class (C06) and by (3.15%) for the class (C23) which resulted in the values (65.01%, 58.47%) respectively. The strategy (Partial+Best+Ids) increased the F1-measure by (1.59%) which resulted in the value (63.98%) for (C20).

The strategy (Complete+All+Ids) increased the F1-measure by (1.93%) which resulted in the value (57.78%) for (C23). The strategy (Complete+Best+Names) increased the F1-measure by (6.37%) for the class (C06) and by (4.28%) for the class (C23) which resulted in the values (65.67%, 59.11%) respectively. The strategy (Complete+Best+Ids) increased the F1-measure by (1.93%) which resulted in the value (64.19%) for (C20).

2.2.2.2 *Analysis*

From previous observations we conclude that maximum increase in F1-Measure (6.37%) was obtained for the class (C06) using the strategy (Complete+Best+Names). In fact, this class is one of the least populated classes and on which we obtained a relatively low F1-Measure (61.74%) using the original corpus. This means that Rocchio using Jaccard (similar to Cosine) did not learn an effective classification model on the original text using the relatively small set of training documents of this class. In addition, Jaccard may not detect enough common features between the classification model and the new documents related to C06. Thus, text Conceptualization enhanced learning and prediction capabilities of Rocchio with Jaccard.

Previous reported improvements at class level influenced the MacroAveraged F1-Measure with a gain of (1.79%, 0.53%, 0.73%) using strategies (Add+Best+Names), (Partial+Best+Names) and (Complete+Best+Names) respectively. Note that the overall performance of Rocchio using Jaccard on the original corpus is significantly different from its performance on the corpus after applying the strategy (Add+Best+Names) according to McNemar test.

In fact, using names of concepts to enrich text is useful to Rocchio with Jaccard and especially with the names are added into text. Using Ids seems to be less interesting; however it is relatively useful to classes like (C20) which is one of the least populated classes like (C06) and (C23) which is a large class. It seems that models of large classes built using concepts instead of words are more effective.

According to Figure 52, using the strategy (Add+Best+Names) increased F1-Measure of the five classes which improved significantly the overall performance of Rocchio with Jaccard. This improvement is the maximum increase obtained among all other strategies that results in a MacroAveraged F1-measure of (69.56%) as presented formerly in Table 28. Note that for each of the three strategies (Add, Partial, Complete) maximum number of improved classes after conceptualization is obtained using the names of the best mapped concepts. This is due to Rocchio's dependency on text statistics that is fortified using names rather than using Ids.

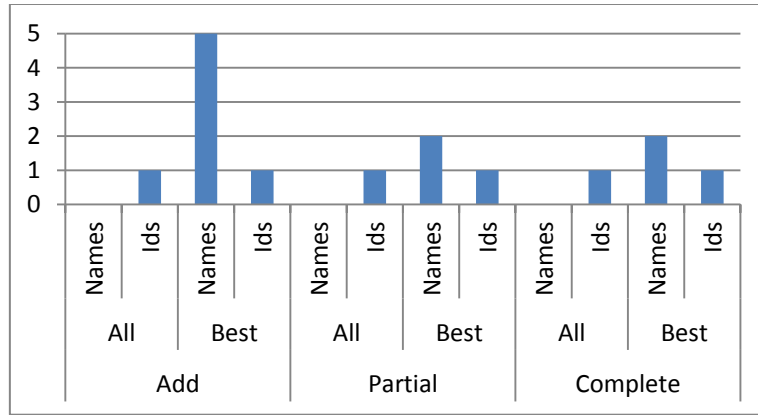


Figure 52. Number of classes with improved F1-Measure on conceptualized text compared with the original text using Rocchio with Jaccard similarity measure

2.2.3 Results using Rocchio with KullbackLeibler

2.2.3.1 Observations

According to results illustrated in Table 29, the F1-measure obtained from applying Rocchio with KullbackLeibler similarity measure on the original Ohsumed corpus varied from (58.53%) to (72.54%) for classes (C23, C14) respectively. Conceptualization resulted in improvements using strategies: (Add+Best+Names), (Add+Best+Ids), (Partial+Best+Names), (Partial+Best+Ids) and (Complete+Best+Ids). All of these improvements appeared at the MacroAveraged level except for (Partial+Best+Names) where results for only two classes (C06 and C20) were improved.

The strategy (Add+Best+Names) improved the performance of Rocchio (except for the class C23) with KullbackLeibler similarity measure by a percentage that varies from (0.37%) for the class (C04) to (1.41%) for the class (C20). The absolute value of F1-measure varied from (58.07%) to (73.27%) for classes (C23, C14) respectively. The strategy (Add+Best+Ids) improved the performance of Rocchio with KullbackLeibler similarity measure by a percentage that varies from (2.91%) for the class (C23) to (7.07%) for the class (C14). The absolute value of F1-measure varied from (60.23%) to (77.66%) for classes (C23, C14) respectively.

The strategy (Partial+Best+Names) improved the performance of Rocchio with KullbackLeibler similarity measure by a percentage of (0.66%) for the class (C06) and (0.16%) for the class (C20). The resulting values of F1-measure are (66.09%) and (63.74%) respectively. The strategy (Partial+Best+Ids) improved the performance of Rocchio with KullbackLeibler similarity measure by the percentages (5.20%, 7.56%, 2.56%) for the classes (C04, C14, C20) resulting in F1-measures of (71.61%, 78.02%, 65.27%) respectively.

The strategy (Complete+Best+Ids) improved the performance of Rocchio (except for the class C23) with KullbackLeibler similarity measure by a percentage that varies from (0.42%) for the class (C06) to (8.00%) for the class (C14). The absolute value of F1-measure varied from (56.51%) to (78.34%) for classes (C23, C14) respectively.

Corpus \ Class			C04		C06		C14		C20		C23		Macro		Micro	
Original			68.07		65.66		72.54		63.64		58.53		65.69		65.53	
Add	All	Names	59.62	-12.41*	56.99	-13.20*	64.32	-11.33*	55.75	-12.39*	53.81	-8.07	58.10	-11.55*	58.05	-11.42
		Ids	62.82	-7.71*	59.88	-8.81	69.12	-4.70*	61.01	-4.13	52.90	-9.62*	61.15	-6.91*	61.00	-6.92
	Best	Names	68.32	+0.37	66.19	+0.81	73.27	+1.02	64.54	+1.41	58.07	-0.79	66.08	+0.60	65.87	+0.52
		Ids	72.72	+6.83*	69.72	+6.19*	77.66	+7.07*	66.58	+4.62*	60.23	+2.91*	69.38	+5.63*	69.48	+6.03
Partial	All	Names	56.87	-16.45*	54.34	-17.24*	62.23	-14.21*	53.29	-16.27*	51.85	-11.41*	55.72	-15.18*	55.70	-15.00
		Ids	59.72	-12.27*	58.89	-10.31	67.66	-6.73*	58.33	-8.35*	51.92	-11.29*	59.30	-9.72*	59.11	-9.80
	Best	Names	66.84	-1.81	66.09	+0.66	72.32	-0.30	63.74	+0.16	57.40	-1.93*	65.28	-0.62	65.05	-0.73
		Ids	71.61	+5.20*	65.42	-0.37	78.02	+7.56*	65.27	+2.56*	55.91	-4.47*	67.25	+2.37*	67.90	+3.61
Complete	All	Names	56.74	-16.64*	54.60	-16.84*	62.15	-14.32*	53.25	-16.33*	51.99	-11.17*	55.75	-15.13*	55.72	-14.97
		Ids	59.63	-12.40*	58.72	-10.57	67.69	-6.69*	57.88	-9.05*	51.86	-11.39*	59.16	-9.94*	58.99	-9.98
	Best	Names	65.84	-3.27*	65.20	-0.71	72.10	-0.60	62.88	-1.20	57.31	-2.09	64.66	-1.56	64.45	-1.65
		Ids	71.54	+5.09*	65.93	+0.42	78.34	+8.00*	64.07	+0.68*	56.51	-3.45*	67.28	+2.42	67.94	+3.67

Table 29. Results of applying Rocchio with KullbackLeibler similarity measure to Ohsumed corpus and to the results of its conceptualization according to 12 conceptualization strategies. (*) denotes significance according to McNemar. Values in the table are percentages.

2.2.3.2 Analysis

From previous observations we conclude that maximum increase in F1-Measure (8%) was obtained for the class (C14) using the strategy (Complete+Best+Ids). In fact, this class is one of the most populated classes and on which we obtained the highest F1-Measure (72.54%) using the original corpus. It seems that in this case, using Ids in text enrichment helped Rocchio with KullbackLeibler enhance its capability to distinguish classes which depended on the quality of the classification model; highly populated classes are easier to learn than least populated ones so they have more effective classification model.

Previous reported improvements at class level influenced the MacroAveraged F1-Measure with a gain of (0.60%, 5.63%, 2.37%, 2.42%) using strategies (Add+Best+Names), (Add+Best+Ids), (Partial+Best+Ids), (Complete+Best+Ids) respectively. Note that the overall performance of Rocchio using KullbackLeibler on the original corpus is significantly different from its performance on the corpus after applying the strategies (Add+Best+Ids) and (Partial+Best+Ids) according to McNemar test. In fact, using Ids for text enrichment seem to improve the performance of Rocchio using KullbackLeibler as the similarity measure. Having Ids in text forces the indexer to use the entire concepts as a feature, these Ids are more distinctive than words in the space model which is very beneficial to KullbackLeibler. In fact KullbackLeibler is based on the divergence of feature distribution in compared vectors.

According to Figure 53, using the strategy (Add+Best+Ids) increased F1-Measure of the five classes which improved significantly the overall performance of Rocchio with KullbackLeibler. This improvement is the maximum increase obtained among all other strategies that results in a MacroAveraged F1-measure of (69.38%) as presented formerly in Table 29. Note that for each of the three strategies (Add, Partial, Complete) maximum number

of improved classes after conceptualization is obtained using the identifiers of the best mapped concepts.

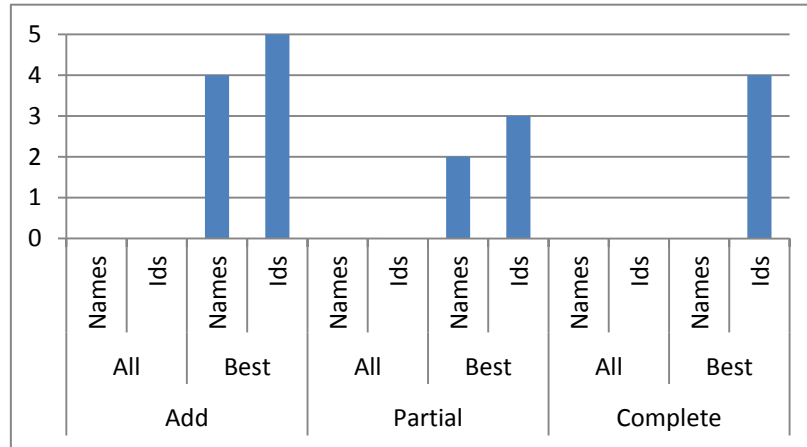


Figure 53. Number of classes with improved F1-Measure on conceptualized text compared with the original text using Rocchio with KullbackLeibler similarity measure

2.2.4 Results using Rocchio with Levenshtein

2.2.4.1 Observations

According to results illustrated in Table 30, the F1-measure obtained from applying Rocchio with Levenshtein similarity measure on the original Ohsumed corpus varied from (53.87%) to (78.67%) for classes (C23, C14) respectively. All strategies using Names of mappings in addition to (Add+Best+Ids) improved classification results. Note that the most effective conceptualization strategy is (Add+Best+Names).

The strategy (Add+All+Names) improved the performance of Rocchio with Levenshtein similarity measure by a percentage of (13.85%) for the class (C06) and (9.66%) for the class (C23). The resulting values of F1-measure are (63.48) and (59.07%) respectively. The strategy (Add+Best+Names) improved the performance of Rocchio (except for the class C20) with Levenshtein similarity measure by a percentage that varies from (0.40%) for the class (C14) to (16.14%) for the class (C06). The absolute value of F1-measure varied from (60.90%) to (78.98%) for classes (C23, C14) respectively. The strategy (Add+Best+Ids) increased significantly the F1-measure by (1.01%) which resulted in the value (79.46%) for (C14).

The strategy (Partial+All+Names) increased the F1-measure by a percentage of (13.39%) for the class (C06) and (7.71%) for the class (C23). The resulting values of F1-measure are (63.22%) and (58.02%) respectively. The strategy (Partial+Best+Names) increased the F1-measure by a percentage of (15.12%) for the class (C06) and (12.60%) for the class (C23). The resulting values of F1-measure are (64.19%) and (60.66%) respectively.

The strategy (Complete+All+Names) increased significantly the F1-measure by a percentage of (12.96%) for the class (C06) and (7.68%) for the class (C23). The resulting values of F1-measure are (62.98%) and (58.00%) respectively. The strategy (Complete+Best+Names) increased significantly the F1-measure by a percentage of (18.58%) for the class (C06) and

(14.56%) for the class (C23). The resulting values of F1-measure are (66.12%) and (61.71%) respectively.

Corpus \ Class			C04		C06		C14		C20		C23		Macro		Micro	
Original			77.01		55.76		78.67		64.40		53.87		65.94		66.89	
Add	All	Names	72.68	-5.62*	63.48	+13.85*	74.00	-5.93*	55.20	-14.28*	59.07	+9.66*	64.89	-1.60*	65.65	-1.86
		Ids	73.95	-3.97*	45.94	-17.60	77.62	-1.33*	60.20	-6.53*	35.94	-33.28*	58.73	-10.93*	60.41	-9.69
	Best	Names	77.55	+0.70*	64.76	+16.14*	78.98	+0.40	64.24	-0.24*	60.90	+13.06*	69.29	+5.08*	70.06	+4.74
		Ids	76.02	-1.29*	44.62	-19.97*	79.46	+1.01*	63.57	-1.29	15.27	-71.66*	55.79	-15.40*	59.63	-10.86
Partial	All	Names	70.75	-8.12*	63.22	+13.39*	72.01	-8.46*	52.12	-19.07*	58.02	+7.71*	63.23	-4.12*	63.94	-4.41
		Ids	70.32	-8.69*	49.06	-12.01	75.72	-3.75*	57.98	-9.97*	49.73	-7.68*	60.56	-8.16*	61.28	-8.40
	Best	Names	76.49	-0.67	64.19	+15.12*	77.72	-1.20*	62.69	-2.66*	60.66	+12.60*	68.35	+3.66*	69.12	+3.33
		Ids	67.04	-12.95*	42.65	-23.51*	68.91	-12.40*	51.89	-19.42*	6.95	-87.10*	47.49	-27.98*	51.50	-23.01
Complete	All	Names	70.70	-8.20*	62.98	+12.96*	71.71	-8.84*	52.04	-19.20*	58.00	+7.68*	63.09	-4.33*	63.80	-4.62
		Ids	70.49	-8.46*	50.02	-10.28	75.34	-4.22*	58.17	-9.67*	50.50	-6.26	60.91	-7.63*	61.60	-7.92
	Best	Names	76.05	-1.24	66.12	+18.58*	75.99	-3.40*	62.91	-2.31*	61.71	+14.56*	68.56	+3.97*	69.04	+3.21
		Ids	64.63	-16.08*	44.37	-20.42	67.99	-13.58*	49.00	-23.91*	12.28	-77.21*	47.65	-27.73*	50.70	-24.21

Table 30. Results of applying Rocchio with Levenshtein similarity measure to Ohsumed corpus and to the results of its conceptualization according to 12 conceptualization strategies. (*) denotes significance according to McNemar test. Values in the table are percentages.

2.2.4.2 Analysis

From previous observations we conclude that maximum increase in F1-Measure (18.58%) was obtained for the class (C06) using the strategy (Complete+Best+Names). In fact, this class is one of the least populated classes and on which we obtained a relatively low F1-Measure (55.76%) using the original corpus. This means that Rocchio using Levenshtein did not learn an effective classification model on the original text using the relatively small set of training documents of this class. In addition, Levenshtein may not detect enough common features between the classification model and the new documents related to C06. Thus, text *Conceptualization* enhanced learning and prediction capabilities of Rocchio with Levenshtein.

Previous reported improvements at class level influenced the MacroAveraged F1-Measure with a gain of (5.08%, 3.66%, 3.97%) using strategies (Add+Best+Names), (Partial+Best+Names), (Complete+Best+Names) respectively. Note that the overall performance of Rocchio using Levenshtein on the original corpus is significantly different from its performance on the corpus after applying either strategy according to McNemar test.

In fact, using names of concepts to enrich text is useful to Rocchio with Levenshtein and especially when the names are added into text. In fact, Rocchio with Levenshtein seems to be highly dependent on text statistics and thus replacing text with Ids of corresponding concepts disturbs learning and classification and results in deterioration in its effectiveness.

According to Figure 54, using the strategy (Add+Best+Names) increased F1-Measure of four classes which improved significantly the overall performance of Rocchio with

Levenshtein. This improvement is the maximum increase obtained among all other strategies that results in a MacroAveraged F1-measure of (69.29%) as presented formerly in Table 30. Note that for each of the three strategies (Add, Partial, Complete) maximum number of improved classes after conceptualization is obtained using the names of the best mapped concepts. This is due to Rocchio's dependency on text statistics that is fortified using names rather than using Ids.

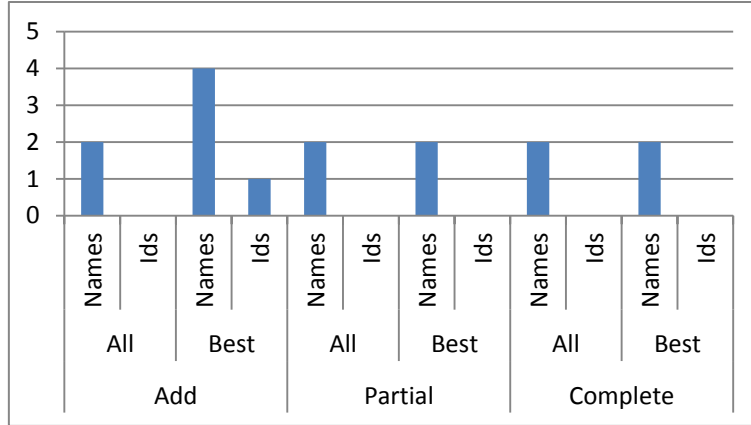


Figure 54. Number of classes with improved F1-Measure on conceptualized text compared with the original text using Rocchio with Levenshtein similarity measure

2.2.5 Results using Rocchio with Pearson

2.2.5.1 Observations

Corpus \ Class			C04		C06		C14		C20		C23		Macro		Micro	
Original			77.87		67.13		80.34		63.13		61.08		69.91		70.85	
Add	All	Names	72.03	-7.51*	62.08	-7.52	75.99	-5.42*	57.33	-9.19*	57.10	-6.52*	64.90	-7.16*	65.99	-6.85
		Ids	74.21	-4.70*	62.96	-6.21*	76.92	-4.26*	62.49	-1.02*	59.64	-2.36	67.24	-3.82*	68.04	-3.96
	Best	Names	77.75	-0.16	68.73	+2.39*	79.97	-0.47	63.12	-0.03	61.34	+0.44*	70.18	+0.39	70.91	+0.08
		Ids	77.42	-0.58	66.41	-1.07	79.46	-1.09*	64.15	+1.61	61.15	+0.13*	69.72	-0.27	70.55	-0.42
Partial	All	Names	70.56	-9.40*	58.87	-12.31*	74.67	-7.06*	54.07	-14.35*	55.22	-9.59*	62.68	-10.35*	64.02	-9.63
		Ids	73.26	-5.93*	61.43	-8.48*	76.31	-5.01*	61.99	-1.82*	59.10	-3.23	66.42	-4.99*	67.26	-5.07
	Best	Names	76.27	-2.06	66.83	-0.44*	78.94	-1.75*	60.49	-4.18*	60.62	-0.74*	68.63	-1.83*	69.52	-1.87
		Ids	71.46	-8.24*	54.23	-19.21*	70.47	-12.29*	62.24	-1.41*	54.60	-10.61*	62.60	-10.46*	63.44	-10.45
Complete	All	Names	70.48	-9.50*	58.87	-12.31*	74.78	-6.93*	54.07	-14.35*	55.26	-9.52*	62.69	-10.33*	64.04	-9.60
		Ids	73.17	-6.03*	61.49	-8.39*	76.42	-4.88*	61.99	-1.82*	59.19	-3.09	66.45	-4.94*	67.30	-5.01
	Best	Names	76.76	-1.43	67.05	-0.12*	78.84	-1.87*	61.06	-3.29*	60.44	-1.05*	68.83	-1.55*	69.68	-1.64
		Ids	71.77	-7.83*	53.91	-19.69*	70.57	-12.17*	61.93	-1.90*	54.24	-11.19*	62.48	-10.62*	63.38	-10.54

Table 31. Results of applying Rocchio with Pearson similarity measure to Ohsumed corpus and to the results of its conceptualization according to 12 conceptualization strategies. (*) denotes significance according to McNemar test. Values in the table are percentages.

According to results illustrated in Table 31, the F1-measure obtained from applying Rocchio with Pearson similarity measure on the original Ohsumed corpus varied from (61.08%) to (80.34%) for classes (C23, C14) respectively. Only two conceptualization strategies improved classification: (Add+Best+Names) and (Add+Best+Ids).

The strategy (Add+Best+Names) increased the F1-measure by a percentage of (2.39%) for the class (C06) and (0.44%) for the class (C23). The resulting values of F1-measure are (68.73%) and (61.34%) respectively.

The strategy (Add+Best+Ids) increased the F1-measure by a percentage of (1.61%) for the class (C20) and (0.13%) for the class (C23). The resulting values of F1-measure are (64.15%) and (61.15%) respectively.

2.2.5.2 Analysis

From previous observations we conclude that maximum increase in F1-Measure (2.39%) was obtained for the class (C06) using the strategy (Add+Best+Names). In fact, this class is one of the least populated classes. This is similar to our observations on Cosine which is logical as Pearson is a modified Cosine similarity measure using centered vectors.

Previous reported improvements at class level influenced the MacroAveraged F1-Measure with a gain of (0.39%) using strategies (Add+Best+Names). Thus, using Names of concepts in text enrichment can improve text classification using Rocchio with Pearson (similar to Cosine). Note that there no evidence that this improvement is significant according to McNemar test.

According to Figure 55, using the strategy (Add+Best+Names) or (Add+Best+Ids) increased F1-Measure of two classes. The improvement using the first one is the maximum increase obtained among all other strategies that results in a MacroAveraged F1-measure of (70.18%) as presented formerly in Table 31.

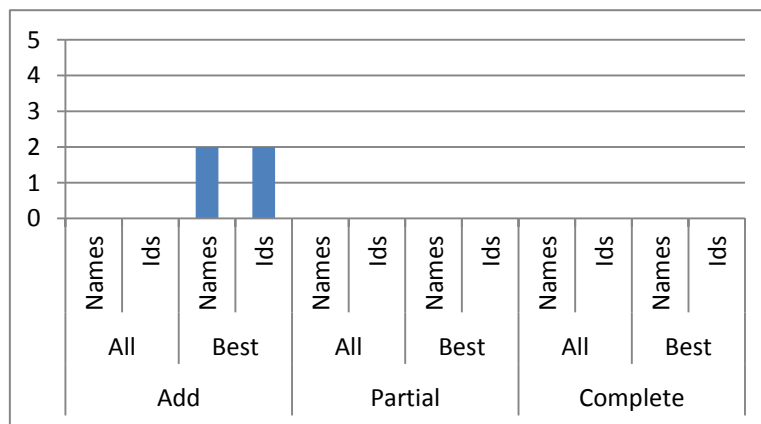


Figure 55. Number of classes with improved F1-Measure on conceptualized text compared with the original text using Rocchio with Pearson similarity measure

2.2.6 Results using NB

2.2.6.1 Observations

According to results illustrated in Table 32, the F1-measure obtained from applying NB on the original Ohsumed corpus varied from (40.90%) to (76.40%) for classes (C23, C14) respectively. All conceptualization strategies improved classification except for three: (Partial+Best+Names), (Partial+Best+Ids) and (Complete+Best+Ids).

The strategy (Add+All+Names) improved the performance of NB with a percentage that varies from (0.92%) for the class (C14) to (13.20%) for the class (C23). The absolute value of F1-measure varied from (46.30%) to (77.10%) for classes (C23, C14) respectively. The strategy (Add+All+Ids) improved the performance of NB with a percentage that varies from (1.44%) for the class (C14) to (27.38%) for the class (C23). The absolute value of F1-measure varied from (46.30%) to (77.10%) for classes (C23, C14) respectively. The strategy (Add+Best+Names) improved the performance of NB (except for C04) with a percentage that varies from (1.05%) for the class (C14) to (7.16%) for the class (C06). The absolute value of F1-measure varied from (41.90%) to (77.20%) for classes (C23, C14) respectively. The strategy (Add+Best+Ids) improved the performance of NB (except for C14) with a percentage that varies from (2.42%) for the class (C04) to (19.32%) for the class (C23). The absolute value of F1-measure varied from (48.80%) to (76.20%) for classes (C23, C14) respectively.

Corpus \ Class			C04		C06		C14		C20		C23		Macro		Micro	
Original			70.30		55.90		76.40		56.30		40.90		59.96		61.30	
Add	All	Names	71.00	+1.00*	61.50	+10.02*	77.10	+0.92*	58.00	+3.02*	46.30	+13.20*	62.78	+4.70*	63.90	+4.24
		Ids	72.70	+3.41*	64.00	+14.49	77.50	+1.44	61.80	+9.77	52.10	+27.38*	65.62	+9.44*	66.60	+8.65
	Best	Names	69.80	-0.71	59.90	+7.16*	77.20	+1.05*	58.30	+3.55	41.90	+2.44	61.42	+2.43*	62.40	+1.79
		Ids	72.00	+2.42*	60.30	+7.87	76.20	-0.26*	57.90	+2.84*	48.80	+19.32*	63.04	+5.14*	64.30	+4.89
Partial	All	Names	71.20	+1.28	62.20	+11.27*	77.40	+1.31	56.90	+1.07	45.10	+10.27*	62.56	+4.34*	63.70	+3.92
		Ids	71.20	+1.28	62.20	+11.27	77.60	+1.57	57.30	+1.78	48.10	+17.60*	63.28	+5.54*	64.50	+5.22
	Best	Names	68.30	-2.84*	55.50	-0.72	75.70	-0.92	55.80	-0.89	39.30	-3.91	58.92	-1.73	60.10	-1.96
		Ids	60.40	-14.08*	47.30	-15.38	68.70	-10.08*	50.00	-11.19	35.10	-14.18*	52.30	-12.78*	53.60	-12.56
Complete	All	Names	69.90	-0.57	63.40	+13.42*	77.60	+1.57	56.60	+0.53	45.10	+10.27*	62.52	+4.27*	63.50	+3.59
		Ids	71.70	+1.99*	61.10	+9.30	77.30	+1.18	57.40	+1.95	47.80	+16.87*	63.06	+5.17*	64.30	+4.89
	Best	Names	68.60	-2.42*	58.40	+4.47*	78.20	+2.36*	57.40	+1.95	37.70	-7.82*	60.06	+0.17	61.10	-0.33
		Ids	62.80	-10.67*	50.70	-9.30	72.40	-5.24*	51.60	-8.35	36.60	-10.51	54.82	-8.57*	56.10	-8.48

Table 32. Results of applying NB to Ohsumed corpus and to the results of its conceptualization according to 12 conceptualization strategies. (*) denotes significance according to McNemar test. Values in the table are percentages.

The strategy (Partial+All+Names) improved the performance of NB with a percentage that varies from (1.07%) for the class (C20) to (11.27%) for the class (C06). The absolute value of

F1-measure varied from (45.10%) to (77.40%) for classes (C23, C14) respectively. The strategy (Partial+All+Ids) improved the performance of NB with a percentage that varies from (1.28%) for the class (C04) to (17.60%) for the class (C23). The absolute value of F1-measure varied from (48.10%) to (77.60%) for classes (C23, C14) respectively.

The strategy (Complete+All+Names) improved the performance of NB (except for C04) with a percentage that varies from (0.53%) for the class (C20) to (13.42%) for the class (C06). The absolute value of F1-measure varied from (45.10%) to (77.60%) for classes (C23, C14) respectively. The strategy (Complete+All+Ids) improved the performance of NB with a percentage that varies from (1.18%) for the class (C14) to (16.87%) for the class (C23). The absolute value of F1-measure varied from (47.80%) to (77.30%) for classes (C23, C14) respectively. The strategy (Complete+Best+Names) improved the performance of NB on three classes (C06, C14, C20) with percentages (4.47%, 2.36%, 1.95%) and absolute values (58.40%, 78.20%, 57.40%) respectively.

2.2.6.2 Analysis

From previous observations we conclude that maximum increase in F1-Measure (27.38%) was obtained for the class (C23) using the strategy (Add+All+Ids). In fact, on this class we obtained the lowest F1-Measure (40.90%) using the original corpus. (C23) is a large class that is usually difficult to distinguish due to the numerous features that it might share with other classes in the feature space. It seems that using Ids of mappings that are more distinctive features than words helps NB to delimit this class and led to this improvement.

Previous reported improvements at class level influenced the MacroAveraged F1-Measure with a gain of (4.70%, 9.44%, 2.43%, 5.14%, 4.34%, 5.54%, 4.27%, 5.17%, 0.17%, 4.27%, 5.17%, 0.17%) using strategies (Add+All+Names), (Add+All+Ids), (Add+Best+Names), (Add+Best+Ids), (Partial+All+Names), (Partial+All+Ids), (Complete+All+Names), (Complete+All+Ids), and (Complete+Best+Names) respectively. Note that the overall performance of NB on the original corpus is significantly different from its performance on the corpus after applying either strategy, except for (Complete+Best+Names), according to McNemar test.

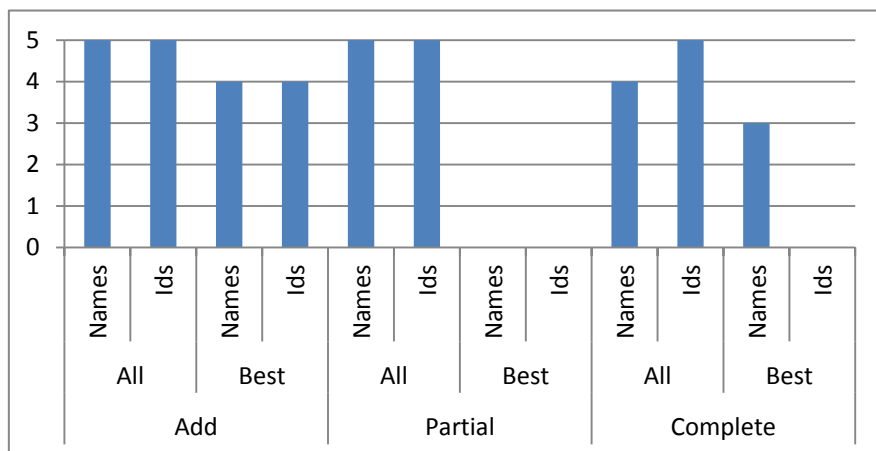


Figure 56. Number of classes with improved F1-Measure on conceptualized text compared with the original text using NB

In fact, most conceptualization strategies improved text classification and particularly those using Ids in enrichment. Using Names is less effective. The most effective conceptualization strategy is (Add+All+Ids). In fact, Ids are more distinctive features than words and it seems that introducing them in the feature space enhanced NB's capabilities to learn the classification model and to predict classes.

According to Figure 56, using the strategy (Add+All+Names), (Add+All+Ids), (Partial+All+Names), (Partial+All+Ids), and (Complete+All+Ids) increased F1-Measure of five classes which improved significantly the overall performance of NB. The maximum overall improvement is obtained using (Add+All+Ids), this results in a MacroAveraged F1-measure of (65.62%) as presented formerly in Table 32. Note that for each of the three strategies (Add, Partial, Complete) maximum number of improved classes after conceptualization is obtained using the Ids of all mapped concepts which is due to its distinctive nature.

2.2.7 Results using SVM

2.2.7.1 Observations

According to results illustrated in Table 33. The F1-measure obtained from applying SVM on the original Ohsumed corpus varied from (56.80%) to (83.00%) for classes (C06, C14) respectively. Most conceptualization strategies increased F1-Measure except for Partial and Complete ones using information of Best mappings.

The strategy (Add+All+Names) improved the performance of SVM (except for the class C14) with a percentage that varies from (0.13%) for the class (C04) to (18.13%) for the class (C06). The absolute value of F1-measure varied from (65.30%) to (82.80%) for classes (C23, C14) respectively. The strategy (Add+All+Ids) improved the performance of SVM with a percentage that varies from (0.60%) for the class (C14) to (26.23%) for the class (C06). The absolute value of F1-measure varied from (67.00%) to (83.50%) for classes (C23, C14) respectively. The strategy (Add+Best+Names) improved the performance of SVM (except for C04) with a percentage that varies from (0.12%) for the class (C14) to (2.99%) for the class (C06). The absolute value of F1-measure varied from (58.50%) to (83.10%) for classes (C06, C14) respectively. The strategy (Add+Best+Ids) improved the performance of SVM (except for C14) with a percentage that varies from (0.25%) for the class (C04) to (10.21%) for the class (C06). The absolute value of F1-measure varied from (62.60%) to (82.90%) for classes (C06, C14) respectively. The strategy (Partial+All+Names) improved the performance of SVM with a percentage that varies from (0.13%) for the class (C04) to (19.72%) for the class (C06). The absolute value of F1-measure varied from (65.40%) to (83.20%) for classes (C23, C14) respectively. The strategy (Partial+All+Ids) improved the performance of SVM on classes (C06, C20, C23) with the corresponding percentages (27.11%, 8.80%, 3.62%). The absolute value of F1-measure varied from (65.90%) to (82.80%) for classes (C23, C14) respectively.

The strategy (Complete+All+Names) improved the performance of SVM (except for C04) with a percentage that varies from (0.24%) for the class (C14) to (19.19%) for the class (C06). The absolute value of F1-measure varied from (65.10%) to (83.20%) for classes (C23, C14) respectively. The strategy (Complete+All+Ids) improved the performance of SVM on

classes (C06, C20, C23) with the corresponding percentages (25.00%, 8.33%, 4.09%). The absolute value of F1-measure varied from (66.20%) to (82.60%) for classes (C23, C14) respectively.

Corpus \ Class			C04		C06		C14		C20		C23		Macro		Micro	
Original			79.90		56.80		83.00		64.80		63.60		69.62		71.90	
Add	All	Names	80.00	+0.13	67.10	+18.13*	82.80	-0.24	68.20	+5.25*	65.30	+2.67*	72.68	+4.40*	74.10	+3.06
		Ids	80.80	+1.13	71.70	+26.23*	83.50	+0.60	73.70	+13.73*	67.00	+5.35*	75.34	+8.22*	76.20	+5.98
	Best	Names	79.80	-0.13	58.50	+2.99*	83.10	+0.12	65.10	+0.46	63.90	+0.47	70.08	+0.66	72.20	+0.42
		Ids	80.10	+0.25	62.60	+10.21*	82.90	-0.12	69.80	+7.72*	65.20	+2.52	72.12	+3.59*	73.70	+2.50
Partial	All	Names	80.00	+0.13	68.00	+19.72*	83.20	+0.24	67.50	+4.17*	65.40	+2.83*	72.82	+4.60*	74.20	+3.20
		Ids	79.90	+0.00	72.20	+27.11*	82.80	-0.24	70.50	+8.80*	65.90	+3.62*	74.26	+6.66*	75.20	+4.59
	Best	Names	77.70	-2.75	51.20	-9.86*	80.90	-2.53*	56.20	-13.27*	60.50	-4.87	65.30	-6.21*	68.20	-5.15
		Ids	72.70	-9.01*	7.30	-87.15*	74.80	-9.88*	43.10	-33.49*	54.60	-14.15	50.50	-27.46*	56.60	-21.28
Complete	All	Names	79.10	-1.00	67.70	+19.19*	83.20	+0.24	67.60	+4.32*	65.10	+2.36*	72.54	+4.19*	73.90	+2.78
		Ids	79.80	-0.13	71.00	+25.00*	82.60	-0.48	70.20	+8.33*	66.20	+4.09*	73.96	+6.23*	75.00	+4.31
	Best	Names	77.30	-3.25*	48.40	-14.79*	81.20	-2.17*	54.40	-16.05*	60.50	-4.87*	64.36	-7.56*	67.70	-5.84
		Ids	73.10	-8.51*	4.40	-92.25*	77.50	-6.63*	40.20	-37.96*	54.30	-14.62	49.90	-28.33*	56.50	-21.42

Table 33. Results of applying SVM to Ohsumed corpus and to the results of its conceptualization according to 12 conceptualization strategies. (*) denotes significance according to McNemar. Values in the table are percentages.

2.2.7.2 Analysis

From previous observations we conclude that maximum increase in F1-Measure (27.11%) was obtained for the class (C06) using the strategy (Partial+All+Ids). In fact, this class is one of the least populated classes and on which we obtained a relatively low F1-Measure (56.80%) using the original corpus. It seems that in this case, using Ids in text enrichment helped SVM enhance its capability to distinguish classes, this usually depends on the quality of the classification model; highly populated classes are easier to learn than least populated.

Previous reported improvements at class level influenced the MacroAveraged F1-Measure with a gain of (4.40%, 8.22%, 0.66%, 3.59%, 4.60%, 6.66%, 4.19%, 6.23%) using strategies (Add+All+Names), (Add+All+Ids), (Add+Best+Names), (Add+Best+Ids), (Partial+All+Names), (Partial+All+Ids) and (Complete+All+Names), (Complete+All+Ids) respectively. Note that the overall performance of SVM on the original corpus is significantly different from its performance on the corpus after applying either strategy, except for (Add+Best+Names), according to McNemar test.

In fact, most conceptualization strategies improved text classification and particularly those using Ids in enrichment. Using Names is less effective. The most effective conceptualization strategy is (Add+All+Ids).

According to Figure 57, using the strategies (Add+All+Ids) and (Partial+All+Names) increased F1-Measure of five classes which improved significantly the overall performance of

SVM. The maximum overall improvement is obtained using (Add+All+Ids) that results in a MacroAveraged F1-measure of (75.34%) as presented formerly in Table 33. Note that for each of the strategies (Partial, Complete) maximum number of improved classes after conceptualization is obtained using the names of all mapped concepts. Whereas using Add conceptualization improved classification when all mapped concepts' Ids are added to text.

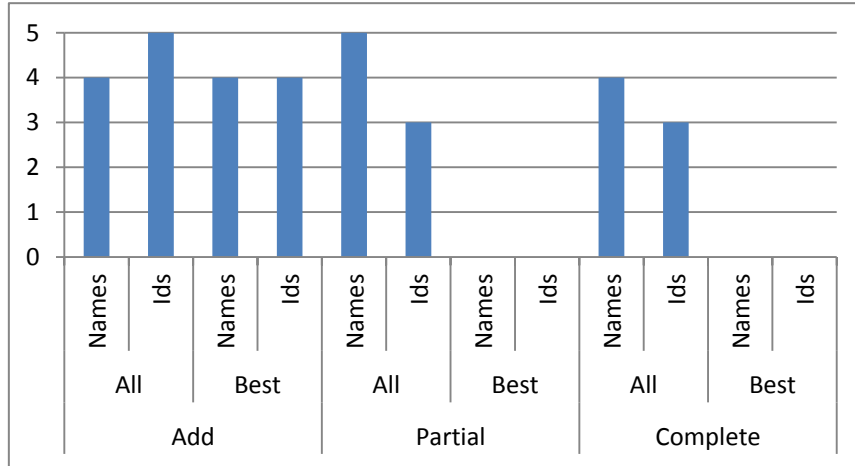


Figure 57. Number of classes with improved F1-Measure on conceptualized text compared with the original text using SVM

2.2.8 Comparing MacroAveraged F1-Measure of the Classification Techniques

After a detailed evaluation of classification results for the seven tested techniques, here we compare their MacroAveraged F1-measure using the textual and the conceptualized corpora. We choose to use the MacroAveraging to avoid penalizing the least populated classes since Ohsumed's classes differ in size substantially. In fact, Table 34 presents these results in columns: one for each technique of classification. Maximum value of F1-measure (75.34%) occurred when testing SVM on the conceptualized Ohsumed according to the strategy (Add+All+Ids). This absolute value and the increase in F1-Measure are quite higher than the values reported in (Bloehdorn *et al.*, 2006; Bai *et al.*, 2010). Note that authors used other classification techniques and/or other subsets of the corpus Ohsumed, so it is difficult to compare their experimental results to ours without harmonizing all details and configurations of both testbeds.

Concerning conceptualization strategies, (Add+Best+Names) is the only strategy that demonstrated improvements for the seven classifiers. A significant increase occurred in F1-measure using Rocchio with three different similarity measures: Cosine, Jaccard and Levenshtein and also using NB according to t-test. In fact, this strategy adds the names of the best mapped concepts to the original text which increases the frequencies of the added words in the text. This has its impact on the index building procedure and helps classifiers emphasize on words that are related to UMLS concepts from the medical domain.

Surprisingly, the strategy (Add+All+Names) caused decreases in F1-measure for all Rocchio-based classifiers and these decreases were significant (except for Levenshtein). This implies that adding all names to the original text add much noise to the feature space that is

difficult to manage by Rocchio-based classifiers. On the contrary, this strategy is beneficial to both NB and SVM. In fact, both techniques use Ids as distinctive feature rather than noise and deploy them to delimit and distinguish classes. These observations apply also to the strategy (Add+All+Ids). Notice that the maximum values of MacroAveraged F1-measure occurred using this strategy for both SVM and NB.

Configuration\Classifier			Rocchio					NB	SVM
			Cosine	Jaccard	Kullback	Levenshtein	Pearson		
Original Text Corpus			69.93	68.33	65.69	65.94	69.91	59.96	69.62
Add	All	Names	64.88 *	64.96*	58.10*	64.89	64.90*	62.78*	72.68
		Ids	67.24 *	66.52*	61.15*	58.73*	67.24*	65.62*	75.34*
	Best	Names	70.69*	69.56*	66.08	69.29*	70.18	61.42*	70.08
		Ids	69.95	67.68	69.38*	55.79	69.72	63.04*	72.12*
Partial	All	Names	62.85 *	63.23*	55.72*	63.23	62.68*	62.56*	72.82
		Ids	66.56 *	66.32*	59.30*	60.56*	66.42*	63.28*	74.26
	Best	Names	69.14	68.70	65.28	68.35	68.63*	58.92*	65.30*
		Ids	62.32 *	60.67*	67.25	47.49*	62.60*	52.30*	50.50*
Complete	All	Names	62.81 *	63.21*	55.75*	63.09	62.69*	62.52*	72.54
		Ids	66.67 *	66.37*	59.16*	60.91*	66.45*	63.06*	73.96
	Best	Names	69.17	68.83	64.66*	68.56	68.83*	60.06	64.36*
		Ids	62.46 *	61.09*	67.28	47.65*	62.48*	54.82*	49.90*

Table 34. MacroAveraged F1-Measure for 7 Classification techniques applied to the original Ohsumed corpus and to the results of its conceptualization according to 12 conceptualization strategies. (*) denotes significance according to t-test (Yang *et al.*, 1999). Values in the table are percentages.

The strategy (Add+All+Ids) increased significantly the F1-measure of both NB and SVM. In fact adding Ids of all the mapped concept to the text according to this strategy implies some impacts on the indexing procedure as Ids are considered as entire tokens and underlying words in the mapped text are not related to them and treated separately. In other words, underlying words are hidden behind the identifier of the concept to which these words are mapped in UMLS, this makes the indexer treat them as different features. In fact, this strategy adds new features from the medical domain to the feature space which helps both classifiers NB and SVM improve their classification. On the contrary, applying Rocchio-based classifiers to conceptualized text according to this strategy decreased F1-measure significantly. This degradation might be principally related to the Ids integrated in text; enriching text with Ids introduced new features in the feature space which affected these classifiers negatively and disturbed their learning and prediction capabilities.

Concerning different classification techniques, we observed improvements in MacroAveraged F1-measure for Rocchio using both Jaccard and Levenshtein after using the names of best mapped concepts (Best+Names) in enriching text according to either “Add”,

“Partial” or “Complete” strategies. As for Cosine and Pearson, improvements were relatively minor and occurred when adding names of best mapped concepts into text.

On the other hand, the effect of conceptualization on Rocchio with KullbackLeibler was different compared to other Rocchio-based classifiers. In fact, we obtained maximum increase in F1-measure using the strategy (Add+Best+Ids). The reason for this difference is that KullbackLeibler similarity measure considers the divergence between feature distributions among documents; it considers “Ids” in classification as useful features compared to others that consider them as noise and prefer using words that constitute “Names” instead.

SVM and NB showed similar behaviors that differ from all Rocchio-based techniques. They prefer “All” to “Best”; they can manage the extra features and prefer conceptualized text using all the mapped concepts from UMLS. Furthermore, they prefer “Ids” to “Names”; they consider the identifiers as useful features and prefer enriching text using them instead of the words that constitute mapped concept names.

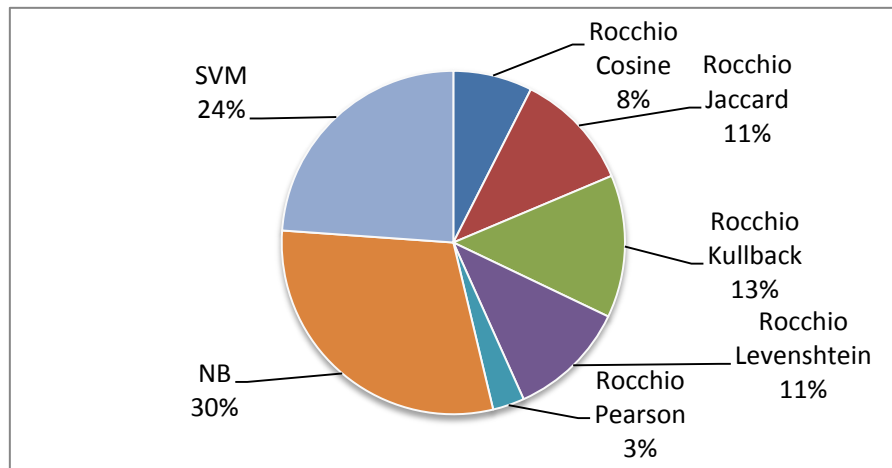


Figure 58. Percentage of Share of each classification technique on the total number of cases where an increase in F1-measure occurred. Cases are gathered from former sections

Figure 58 illustrates the percentage of share of each classification techniques on the total number of cases where we observed an increase in F1-measure for each of the treated classes after applying each of the twelve conceptualization strategies. In fact, NB has the maximum participation with (30%) and SVM comes in the second place with 24%. KullbackLeibler comes first among the Rocchio-based classifiers with (13%). We obtained the same percentage of (11%) for Rocchio with Jaccard and Levenshtein. Rocchio with Cosine improved (8%) of cases and the least percentage is for Rocchio with Pearson that had a share of (3%) only. These results support our observations on the absolute values of F1-measures reported in previous sections.

2.2.9 Comparing F1-Measure of the Classification Techniques for each class

In this section we compare the results from a different point of view. We investigate if conceptualization’s effects on text classification results differ from one class to another. We reported earlier the maximum increases in F1-measure among classes for each conceptualization strategy. According to values in Table 35, Minimum values of F1-measure occurred for the class (C23) and maximum values occurred for the class (C14). This applies for all classification

techniques when applied to the original text and to the conceptualized according to most conceptualization strategies. In fact, (C14) is the most populated class among the treated ones. Despite the large number of documents related to (C23), this class seems to be difficult to classify even after conceptualization. This is due to its large coverage compared to other classes which makes it difficult to distinguish its documents from others.

Maximum F1-Measure values are (80.80%, 72.20%, 83.50%, 73.70%, 67.00%) for the classes (C04, C06, C14, C20, C23) respectively and all of them occurred using SVM. Concerning (C06), the best performance occurred when SVM is applied on conceptualized text using (Partial+All+Ids). As for the other four classes, the best performance occurred when SVM is applied on conceptualized text using (Add+All+Ids). Note that this strategy gave the maximum MacroAveraged F1-measure as well.

Figure 59 illustrates the number of cases where we observed increases in F1-measure for each treated class. Except for Rocchio with KullbackLeibler, all classifiers seem to focus on (C23) after conceptualization. In fact, all classifiers showed poor performance on this class where we reported minimal values of F1-measure. This proves our concept on the effectiveness of semantics in classifying large classes.

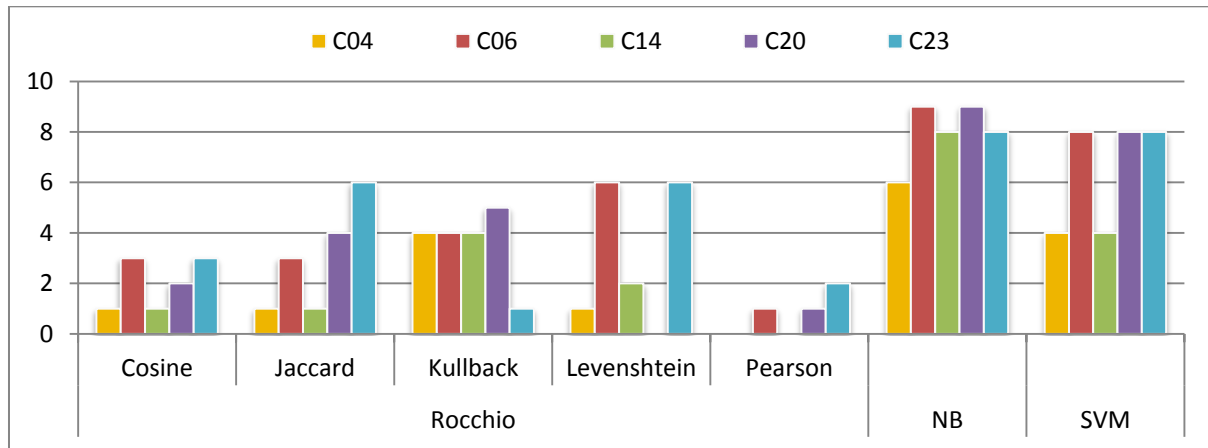


Figure 59. The number of cases where an increase in F1-measure occurred for each class after testing classifiers on all conceptualized versions of Ohsumed.

In addition, the class (C20) and the class (C06) come in the second place after (C23) in the number of improved cases except for Levenshtein. These classes are the least populated classes in the corpus and so all classifiers showed some difficulties in classifying them. This leads us to conclude that integrating semantics helps classifiers overcome learning difficulties on poorly populated classes where the number of example isn't sufficient in order to build an adequate model for the class. In other words, semantics help different techniques to build reliable classification models for a better classification.

System Configuration \ Category				C04	C06	C14	C20	C23	Macro	Micro
Rocchio + Cosine	Original Text Corpus			78.16%	65.92%	81.04%	63.35%	61.17%	69.93%	71.09%
	Add	All	Names	72.06% *	62.05%	76.22% *	57.00% *	57.06%	64.88% *	66.01%
			IDs	74.05% *	62.75%	77.64% *	61.99% *	59.77% *	67.24% *	68.18%
		Best	Names	78.19%	67.99% *	81.17%	64.04%	62.05% *	70.69%	71.63%
			IDs	77.58%	65.27%	80.63% *	64.68%	61.61% *	69.95%	71.01%
	Partial	All	Names	70.35% *	59.27% *	74.87% *	53.90% *	55.84%	62.85% *	64.23%
			IDs	73.28% *	61.44% *	77.21% *	61.59% *	59.29% *	66.56% *	67.54%
		Best	Names	76.78%	66.93% *	79.66% *	60.97% *	61.34% *	69.14%	70.18%
			IDs	71.27% *	53.97% *	70.51% *	62.54% *	53.30% *	62.32% *	63.18%
	Complete	All	Names	70.37% *	59.23% *	74.79% *	53.95% *	55.72%	62.81% *	64.19%
			IDs	73.34% *	61.58% *	77.31% *	61.75% *	59.36% *	66.67% *	67.64%
		Best	Names	76.87%	67.41% *	79.52% *	61.28% *	60.75% *	69.17%	70.12%
			IDs	71.82% *	54.74% *	70.41% *	61.58% *	53.73% *	62.46% *	63.42%
Rocchio + Jaccard	Original Text Corpus			78.23%	61.74%	82.02%	62.98%	56.68%	68.33%	70.12%
	Add	All	Names	72.43% *	61.65% *	77.51% *	57.32% *	55.90% *	64.96% *	66.39%
			IDs	74.95% *	59.15% *	79.18% *	62.46% *	56.85% *	66.52% *	68.02%
		Best	Names	78.83%	64.45%	82.35%	63.69%	58.47% *	69.56% *	71.21%
			IDs	77.96%	59.28%	81.67% *	64.81%	54.66% *	67.68%	69.62%
	Partial	All	Names	71.19% *	58.71% *	75.86% *	55.56% *	54.84% *	63.23% *	64.71%
			IDs	74.09% *	59.21% *	78.71% *	61.98% *	57.61% *	66.32% *	67.70%
		Best	Names	77.34%	65.01% *	80.73% *	61.93% *	58.47% *	68.70%	70.16%
			IDs	71.34% *	51.13% *	71.52% *	63.98% *	45.36% *	60.67% *	62.10%
	Complete	All	Names	71.09% *	58.56% *	75.82% *	55.65% *	54.91% *	63.21% *	64.67%
			IDs	73.99% *	59.28% *	78.88% *	61.92% *	57.78% *	66.37% *	67.74%
		Best	Names	77.26%	65.67% *	80.54% *	61.57% *	59.11% *	68.83%	70.26%
			IDs	71.39% *	52.18% *	70.99% *	64.19% *	46.67% *	61.09% *	62.40%
Rocchio + Kullback	Original Text Corpus			68.07%	65.66%	72.54%	63.64%	58.53%	65.69%	65.53%
	Add	All	Names	59.62% *	56.99% *	64.32% *	55.75% *	53.81%	58.10% *	58.05%
			IDs	62.82% *	59.88%	69.12% *	61.01%	52.90% *	61.15% *	61.00%
		Best	Names	68.32%	66.19%	73.27%	64.54%	58.07%	66.08%	65.87%
			IDs	72.72% *	69.72% *	77.66% *	66.58% *	60.23% *	69.38% *	69.48%
	Partial	All	Names	56.87% *	54.34% *	62.23% *	53.29% *	51.85% *	55.72% *	55.70%
			IDs	59.72% *	58.89%	67.66% *	58.33% *	51.92% *	59.30% *	59.11%
		Best	Names	66.84%	66.09%	72.32%	63.74%	57.40% *	65.28%	65.05%
			IDs	71.61% *	65.42%	78.02% *	65.27% *	55.91% *	67.25% *	67.90%
	Complete	All	Names	56.74% *	54.60% *	62.15% *	53.25% *	51.99% *	55.75% *	55.72%
			IDs	59.63% *	58.72%	67.69% *	57.88% *	51.86% *	59.16% *	58.99%
		Best	Names	65.84% *	65.20%	72.10%	62.88%	57.31%	64.66%	64.45%
			IDs	71.54% *	65.93%	78.34% *	64.07% *	56.51% *	67.28%	67.94%
Rocchio + Levenshtein	Original Text Corpus			77.01%	55.76%	78.67%	64.40%	53.87%	65.94%	66.89%
	Add	All	Names	72.68% *	63.48% *	74.00% *	55.20% *	59.07% *	64.89% *	65.65%
			IDs	73.95% *	45.94%	77.62% *	60.20% *	35.94% *	58.73% *	60.41%
		Best	Names	77.55% *	64.76% *	78.98%	64.24% *	60.90% *	69.29% *	70.06%
			IDs	76.02% *	44.62% *	79.46% *	63.57%	15.27% *	55.79% *	59.63%
	Partial	All	Names	70.75% *	63.22% *	72.01% *	52.12% *	58.02% *	63.23% *	63.94%
			IDs	70.32% *	49.06%	75.72% *	57.98% *	49.73% *	60.56% *	61.28%
		Best	Names	76.49%	64.19% *	77.72% *	62.69% *	60.66% *	68.35% *	69.12%
			IDs	67.04% *	42.65% *	68.91% *	51.89% *	6.95% *	47.49% *	51.50%
	Complete	All	Names	70.70% *	62.98% *	71.71% *	52.04% *	58.00% *	63.09% *	63.80%
			IDs	70.49% *	50.02%	75.34% *	58.17% *	50.50%	60.91% *	61.60%
		Best	Names	76.05%	66.12% *	75.99% *	62.91% *	61.71% *	68.56% *	69.04%
			IDs	64.63% *	44.37%	67.99% *	49.00% *	12.28% *	47.65% *	50.70%

System Configuration \ Category				C04	C06	C14	C20	C23	Macro	Micro
Rocchio + Pearson	Original Text Corpus			77.87%	67.13%	80.34%	63.13%	61.08%	69.91%	70.85%
	Add	All	Names	72.03% *	62.08%	75.99% *	57.33% *	57.10% *	64.90% *	65.99%
			IDs	74.21% *	62.96% *	76.92% *	62.49% *	59.64%	67.24% *	68.04%
		Best	Names	77.75%	68.73% *	79.97%	63.12%	61.34% *	70.18%	70.91%
			IDs	77.42%	66.41%	79.46% *	64.15%	61.15% *	69.72%	70.55%
	Partial	All	Names	70.56% *	58.87% *	74.67% *	54.07% *	55.22% *	62.68% *	64.02%
			IDs	73.26% *	61.43% *	76.31% *	61.99% *	59.10%	66.42% *	67.26%
		Best	Names	76.27%	66.83% *	78.94% *	60.49% *	60.62% *	68.63% *	69.52%
			IDs	71.46% *	54.23% *	70.47% *	62.24% *	54.60% *	62.60% *	63.44%
	Complete	All	Names	70.48% *	58.87% *	74.78% *	54.07% *	55.26% *	62.69% *	64.04%
			IDs	73.17% *	61.49% *	76.42% *	61.99% *	59.19%	66.45% *	67.30%
		Best	Names	76.76%	67.05% *	78.84% *	61.06% *	60.44% *	68.83% *	69.68%
			IDs	71.77% *	53.91% *	70.57% *	61.93% *	54.24% *	62.48% *	63.38%
NB	Original Text Corpus			70.30%	55.90%	76.40%	56.30%	40.90%	59.96%	61.30%
	Add	All	Names	71.00% *	61.50% *	77.10% *	58.00% *	46.30% *	62.78% *	63.90%
			IDs	72.70% *	64.00%	77.50%	61.80%	52.10% *	65.62% *	66.60%
		Best	Names	69.80%	59.90% *	77.20% *	58.30%	41.90%	61.42% *	62.40%
			IDs	72.00% *	60.30%	76.20% *	57.90% *	48.80% *	63.04% *	64.30%
	Partial	All	Names	71.20%	62.20% *	77.40%	56.90%	45.10% *	62.56% *	63.70%
			IDs	71.20%	62.20%	77.60%	57.30%	48.10% *	63.28% *	64.50%
		Best	Names	68.30% *	55.50%	75.70%	55.80%	39.30%	58.92%	60.10%
			IDs	60.40% *	47.30%	68.70% *	50.00%	35.10% *	52.30% *	53.60%
	Complete	All	Names	69.90%	63.40% *	77.60%	56.60%	45.10% *	62.52% *	63.50%
			IDs	71.70% *	61.10%	77.30%	57.40%	47.80% *	63.06% *	64.30%
		Best	Names	68.60% *	58.40% *	78.20% *	57.40%	37.70% *	60.06%	61.10%
			IDs	62.80% *	50.70%	72.40% *	51.60%	36.60%	54.82% *	56.10%
SVM	Original Text Corpus			79.90%	56.80%	83.00%	64.80%	63.60%	69.62%	71.90%
	Add	All	Names	80.00%	67.10% *	82.80%	68.20% *	65.30% *	72.68% *	74.10%
			IDs	80.80%	71.70% *	83.50%	73.70% *	67.00% *	75.34% *	76.20%
		Best	Names	79.80%	58.50% *	83.10%	65.10%	63.90%	70.08%	72.20%
			IDs	80.10%	62.60% *	82.90%	69.80% *	65.20%	72.12% *	73.70%
	Partial	All	Names	80.00%	68.00% *	83.20%	67.50% *	65.40% *	72.82% *	74.20%
			IDs	79.90%	72.20% *	82.80%	70.50% *	65.90% *	74.26% *	75.20%
		Best	Names	77.70%	51.20% *	80.90% *	56.20% *	60.50%	65.30% *	68.20%
			IDs	72.70% *	7.30% *	74.80% *	43.10% *	54.60%	50.50% *	56.60%
	Complete	All	Names	79.10%	67.70% *	83.20%	67.60% *	65.10% *	72.54% *	73.90%
			IDs	79.80%	71.00% *	82.60%	70.20% *	66.20% *	73.96% *	75.00%
		Best	Names	77.30% *	48.40% *	81.20% *	54.40% *	60.50% *	64.36% *	67.70%
			IDs	73.10% *	4.40% *	77.50% *	40.20% *	54.30%	49.90% *	56.50%

Table 35. F1-Measure values for each class using 7 different classifiers and 12 conceptualization strategies. (*) denotes that classifier's performance on the conceptualized Ohsumed is significantly different from its performance on the original Ohsumed according to McNemar test with α equals to (0.05). Increased F1-measure is in bold with a light red background.

2.2.10 Conclusion

According to the results presented in the preceding sections, here we list some remarks. First of all, in most cases, low results are observed when terms are replaced by Ids of their corresponding concepts in the UMLS when using Rocchio-based classifiers except for KullbackLeibler. This performance degradation might be principally related to replacing all terms corresponding to a concept by its Id; only the Ids of concepts can participate in indexing. Terms that are shared among concept with different Ids are excluded from vectors even if they had a high importance. On the contrary, these Ids helped Rocchio with KullbackLeibler, SVM and NB to improve their performance significantly. We presume that they manage these Ids differently from other classifiers and use them as distinctive features rather than noisy ones.

Second, when a Rocchio-based classifier has a good F1-measure value (i.e. exceeds 69%), no significant effect can be observed for the integration of the conceptualization task into the system.

Third, when the system performance using a specific method has a low F1-measure value, as it the case for the classes (C23, C06, and C20), introducing conceptualization can significantly improve this value with a maximum gain reaching (27%) in some cases. Indeed, the class "C23" is very large compared to others and so enriching class representation with semantics might result in a better identification of this class and also in better results. As for (C06, and C20), they have half the number of documents of (C14) which makes learning their classification models more difficult. Conceptualization proved to help overcome this difficulty according to formerly reported results.

Fourth, the best strategy to integrate mapped concept into text is adding them rather than replacing text by concept or keeping only concepts. This implies that mappings retrieved by MetaMap are added into text in order to enrich it with semantics avoiding any information loss and helping classifier in its task by injecting new semantic features into text. Thus we recommend according to our results adding the names of best mapped concepts into text when using Rocchio-based classifiers and the Ids of all mapped concepts when using either NB or SVM.

Finally, it seems useful to introduce domain specific semantic enrichments to classification methods in order to ameliorate their predictions. However, these improvements are relatively dependent on the behavior of the method and also on the used corpus and its class distribution (Albitar, Fournier, *et al.*, 2012b; Albitar, Fournier, *et al.*, 2012a). Consequently, it seems necessary to experimentally define the conditions under which the introduction of semantics can improve classification.

So far, the exploitation of semantic resources was limited in this work. For example, it ignores all relations (like Subsumption and Transversal relations) among concepts that are used in the conceptualization task. Thus, it seems adequate to deploy these relations in the classification process.

3 Experiments applying scenario2 on Ohsumed using Rocchio

In these experiments we intend to enrich text representation after indexing using *Semantic Kernels* in order to assess the impact of this semantic enrichment on text classification applied in the medical domain. Many state of the art works used *Semantic Kernels* with SVM (Bloehdorn *et al.*, 2007; Wang *et al.*, 2008; Aseervatham *et al.*, 2009; Séaghdha, 2009). To the best of our knowledge, this enrichment was not tested with any other classification technique. In this section, the platform implements Rocchio classification technique in order to evaluate its performance after applying *Semantic Kernels* according to the second scenario of the previous chapter. Our choice of Rocchio is for reasons of efficiency and extendibility.

This section presents the platform for our experiments in some details and then presents the results from different points of views and concludes with some recommendation on the use *Semantic Kernels* for text classification using Rocchio.

3.1 Platform for supervised text classification deploying Semantic Kernels

In order to assess the effect of *Semantic Kernels* on the process of text classification using Rocchio, we use the experimental platform illustrated in Figure 60. This platform uses Rocchio for training and prediction as the classification technique. Similar to the previous platform, conceptualization is realized on text before indexing and then the upper part of the figure concerns training phase in which a Rocchio learns the centroids on the enriched index of the conceptualized corpus. Whereas, the lower part illustrates the classification phase in which Rocchio compared the centroids with the enriched index of each new document in order to predicate its class of each test document. This document is represented using the same vocabulary and weighting scheme as those used to represent the training corpus.

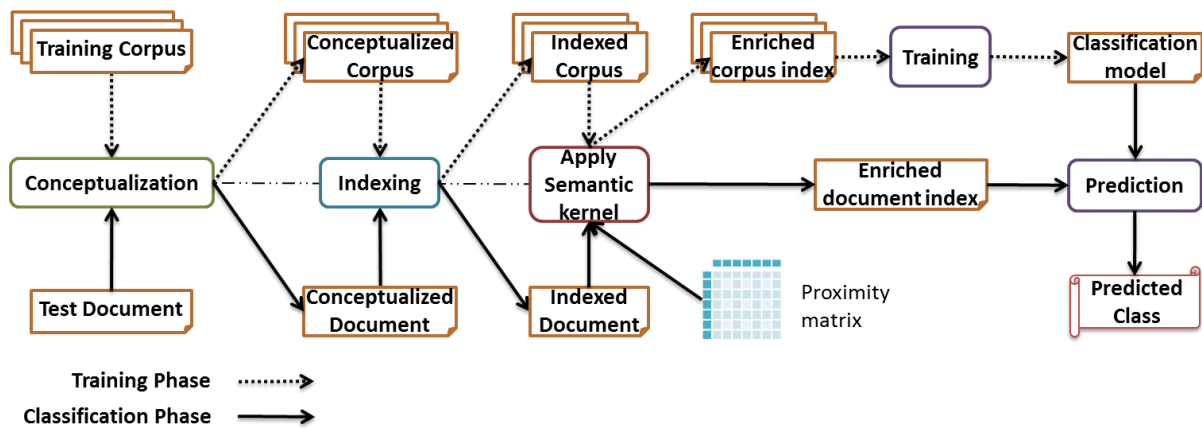


Figure 60. Platform for supervised text classification deploying Semantic Kernels

Next sections present text Conceptualization task, proximity matrix and enriching vectors using *Semantic Kernels* in some details.

3.1.1 Text Conceptualization task

In apply *Semantic Kernels*, we use the complete conceptualization strategy using the Ids of the best mappings (Complete+Best+Ids) (see section 2.1.1). In fact, this strategy guarantees that text will be represented by a BOC after indexing; complete strategy keeps only concepts in text and the Id of each concept is indexed as one feature. Furthermore, for reasons related to efficiency we choose to conceptualize text using concepts from SNOMED-CT exclusively. In fact, using the whole UMLS for assessing semantic similarities is very time consuming. In addition, SNOMED-CT provides a huge knowledge base with large coverage of medical clinical terms (Ruch *et al.*, 2008).

These configurations are used in the platforms of section 4 and section 5 for text conceptualization.

3.1.2 Proximity matrix

The previous chapter introduced proximity matrix and proposed a platform for generating these matrices using UMLS. As we limit the use of UMLS to SNOMED-CT in the rest of this chapter, the semantic similarity engine deploys UMLS::Similarity (McInnes *et al.*, 2009) in order to assess the similarity between SNOMED-CT concepts of the vocabulary pair-to-pair. Resulting similarities are stored in a proximity matrix. Furthermore, in the previous chapter we argued our choice of the five structure-based similarity measures in our experiments. This choice is a compromise between efficiency and effectiveness. A proximity matrix is built for each five similarity measures using the platform resulting in five proximity matrices.

The five chosen semantic similarity measures are cdist, lch, nam, wup and zhong (see chapter 4 section 6.2.2). cdist, wup and zhong generate semantic similarities in the interval [0,1]. Both cdist and zhong return low values whereas wup returns relatively higher values. The measure nam returns similarities in [0, 0.2058] with small variations between different values whereas lch returns similarities in [0, 4.2195] that are the highest absolute values among all other measures. These details are synthesized in Table 36.

Measure	Minimum	Maximum	Observations
cdist	0	1	2 nd lowest values after zhong
lch	0	4.2195	Highest absolute values
nam	0	0.2058	Small variations in values
wup	0	1	Highest values on the scale [0,1]
zhong	0	1	Lowest absolute values

Table 36. Five semantic similarity measures: intervals and observations on their values

In the literature, most works concerning semantic similarity measures use standard datasets in order to evaluate the correlation between each similarity measure and experts ratings. We used a well-known dataset of 30 pairs of concepts from (Pedersen *et al.*, 2012). This dataset, illustrated in Table 37, was annotated by 3 physicians and 9 medical index experts. The annotators gave each pair a note on a 4-point scale corresponding to the following interpretations: practically

synonymous, related, marginally related, and unrelated. The average correlation between physicians is 0.68, and between experts is 0.78. We use in experiments the ratings of experts because they are more than the physicians, and the agreement between them (0.78) is higher than that between physicians (0.68) (Al-Mubaid *et al.*, 2006).

Concept ₁	Concept ₂	Physicians	Experts
Renal failure	Kidney failure	4	4
Abortion	Miscarriage	3	3.3
Heart	Myocardium	3.3	3
Stroke	Infarct	3	2.8
Delusion	Schizophrenia	3	2.2
Calcification	Stenosis	2.7	2
Tumor metastasis	Adenocarcinoma	2.7	1.8
Congestive heart failure	Pulmonary edema	3	1.4
Pulmonary fibrosis	Malignant tumor of lung	1.7	1.4
Diarrhea	Stomach cramps	2.3	1.3
Mitral stenosis	Atrial fibrillation	2.3	1.3
Brain tumor	Intracranial hemorrhage	2	1.3
Antibiotic	Allergy	1.7	1.2
Pulmonary embolus	Myocardial infarction	1.7	1.2
Carpal tunnel syndrome	Osteoarthritis	2	1.1
Rheumatoid arthritis	Lupus	2	1.1
Acne	Syringe	2	1
Diabetes mellitus	Hypertension	2	1
Cortisone	Total knee replacement	1.7	1
Cholangiocarcinoma	Colonoscopy	1.3	1
Lymphoid hyperplasia	Laryngeal cancer	1.3	1
Appendicitis	Osteoporosis	1	1
Depression	Cellulitis	1	1
Hyperlipidemia	Tumor metastasis	1	1
Multiple sclerosis	Psychosis	1	1
Peptic ulcer disease	Myopia	1	1
Rectal polyp	Aorta	1	1
Varicose vein	Entire knee meniscus	1	1
Xerostomia	Alcoholic cirrhosis	1	1

Table 37. A subset of 30 medical concept pairs manually rated by medical experts and physicians for semantic similarity

Using UMLS::Similarity, we evaluated the spearman's correlation coefficient between each of the five chosen similarities (*cdist*, *lch*, *wup*, *nam*, *zhong*) and the ratings of experts and

physicians. The results of tests are illustrated in Table 38. We report maximum correlation between *zhong* and the ratings of experts.

The corpus is composed of 30 pairs of concepts only, which is not sufficiently large and representative of the domain. In addition, the differences between the correlation coefficients are marginal. Nevertheless, we will use these correlation coefficients in results analysis especially those related to experts ratings that are more reliable.

Measure	Physicians	Experts
cdist	0.3116	0.5037
lch	0.3116	0.5037
wup	0.3738	0.5104
nam	0.3329	0.5116
zhong	0.3323	0.5264

Table 38. Spearman’s correlation between five similarity measures and human judgment on Pedersen’s corpus (Pedersen et al., 2012).

3.1.3 Enriching vectors using Semantic Kernels

In this step, we enrich training corpus and each new document using the proximity matrix. Five different proximity matrices are built using each of the semantic similarity measure. Applying the semantic kernel on a document vector ($wc_1 \quad wc_2 \quad wc_3$) is as the following:

$$(wc_1 \quad wc_2 \quad wc_3) * \begin{pmatrix} sim(c_1, c_1) & sim(c_2, c_1) & sim(c_3, c_1) \\ sim(c_1, c_2) & sim(c_2, c_2) & sim(c_3, c_2) \\ sim(c_1, c_3) & sim(c_2, c_3) & sim(c_3, c_3) \end{pmatrix} = (wc_1' \quad wc_2' \quad wc_3') \quad (77)$$

Where:

wc_i is the weight of the concept i in the document

$sim(c_i, c_j)$ is the semantic similarity between two concepts c_i, c_j of the vocabulary $\{c_1, c_2, c_3\}$

Vectors resulting of applying *Semantic Kernel* to the training corpus documents are the input to the training step in order to learn the classification model. Enriched index of test documents are the input to prediction step.

In experiments, the number of the similar concepts involved in text representation after enrichment can be limited. We vary this parameter from 1 to 10 in order to evaluate its effect on the process of classification.

3.2 Evaluating results

In these experiments, the platform executes learning five times once for each of the proximity matrices and once for each value of the parameter related to the number of the similar concepts used in the enrichment. This means that the Rocchio learns $5 * 10 = 50$ different classification models or ensembles of centroids. As for classification, Rocchio uses each of the preceding models with each of its variants (Cosine, Jaccard, KullbackLeibler, Levenshtein, Pearson)

resulting in $5 * 5 * 10 = 250$ executions. The MacroAveraged F1-measure resulting from these executions that are related to each semantic similarity measure are grouped together in the five graphics of Figure 61 in order to analyze the impact of the number of similar concepts used in enrichment on the effectiveness of the five variants of Rocchio.

3.2.1 Observations

This section presents observations on the results that are synthesized in Figure 61. Concerning *cdist*, all variants of Rocchio showed a decrease in F1-measure as soon as the similar concepts are added to text representation. We mention that Cosine, KullbackLeibler and Pearson showed similar behaviors and had the least decrease in F1-Measure from 65% to 32%. We noticed an important decrease in F1-Measure for both Cosine and KullbackLeibler till adding the five most similar concepts. As for Jaccard, the decrease is from 65% to 22% while it varied with Levenshtein from 55% to 12%. Note that most Rocchio variants showed a similar behavior after adding the seventh similar concept.

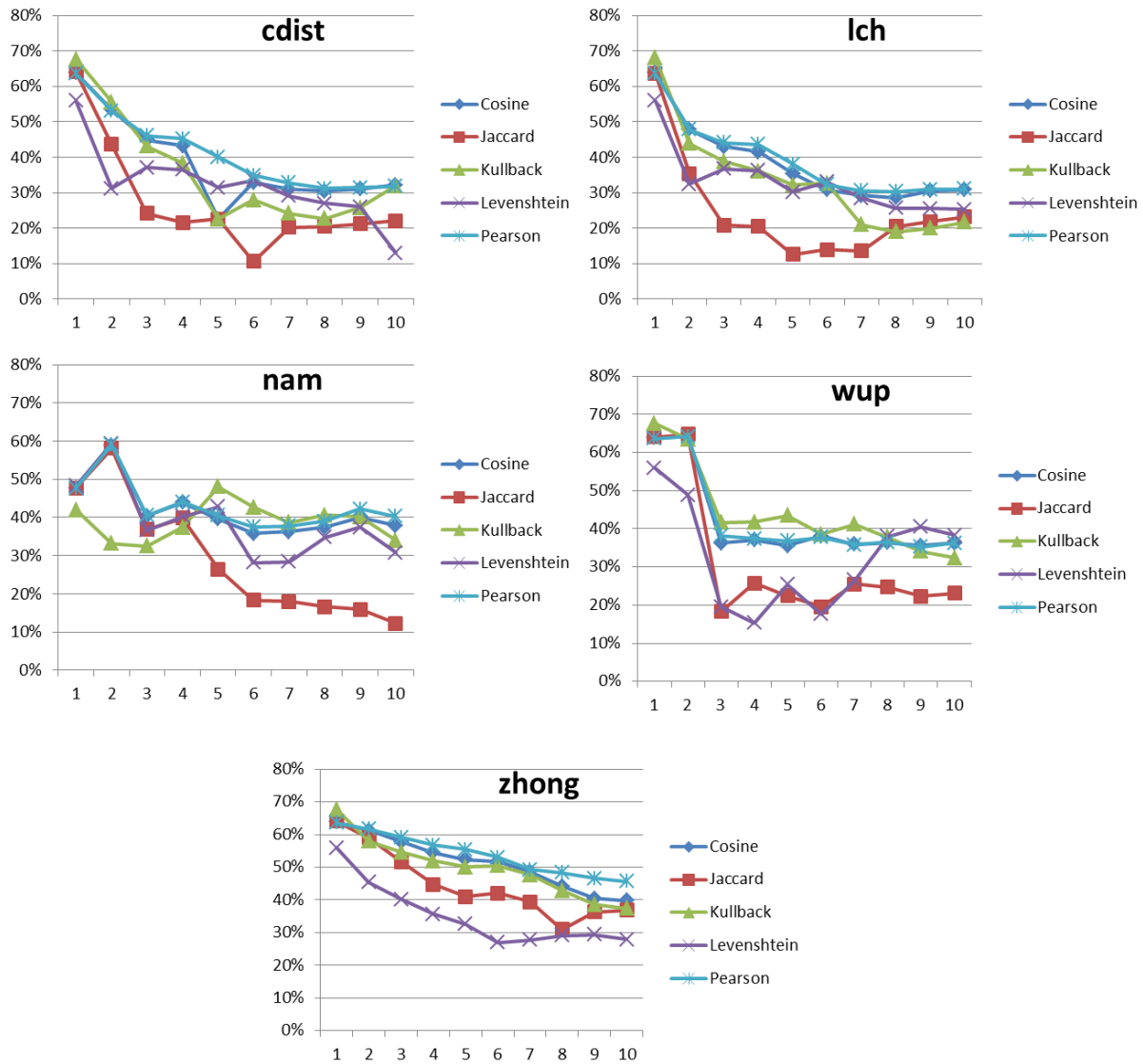


Figure 61. Results of applying Semantic Kernels using *cdist*, *lch*, *nam*, *wup*, *zhong* semantic similarity measures and five variants of Rocchio

Concerning lch and similarly to cdist, all variants of Rocchio showed a decrease in F1-measure as soon as the similar concepts are added to text representation. Note that all of the variants except for Jaccard showed similar behavior. Pearson and cosine had the least decrease in F1-Measure from 65% to 30%. As for Levenshtein, F1-measure decreased from 55% to 25%. KullbackLeibler and Jaccard showed decrease to 22%.

Concerning nam, the decrease in F1-Measure was relatively small except for Jaccard yet the effectiveness is not promising as it decreases as more similar concepts are used in enrichment. Cosine and Pearson showed similar behavior and enriching vectors decreased the F1-Measure from 48% to 38%. Approximately the same decreases are noted for KullbackLeibler and Levenshtein. Maximum decrease in F1-Measure is noted with Jaccard from 48% to 12%.

Concerning wup, introducing similar concepts in text representation caused much decrease in F1-Measure for all variants starting from the three most similar concepts. We report similar behavior for Cosine and Pearson where F1-measure decreased from 65% to 48% whereas F1-Measure decreased from 68% to 32% with KullbackLeibler. Note that less decrease occurred with Levenshtein where F1-Measure varies from 55% to 48%. The maximum deterioration in performance occurred with Jaccard as F1-Measure decreased from 65% to 23%.

Concerning zhong, we note that all variants of Rocchio showed similar behavior and decrease in F1-Measure occurred after adding similar concepts to text representation. The maximum value of F1-Measure varied from 55% to 76% and the minimum varied from 29% to 48%. Pearson demonstrated maximum effectiveness whereas Levenshtein showed the minimum before and after enrichment.

Note that most Rocchio variants using the five semantic similarity measures showed a similar behavior after adding the seventh similar concept. Jaccard showed the worst values of F1-Measures except when using zhong as a semantic similarity measure.

3.2.2 Analysis and conclusion

Previous section presented our observations on the results of classification after representation enrichment using Semantic Kernels. We tested five different variants of Rocchio using five different semantic similarity measures and fixed the number of most similar concepts used in the enrichment from 1 to 10.

According to observations, two variants of Rocchio showed very similar behavior: Cosine and Pearson. In fact, Pearson is considered as a centered Cosine as all vectors are centered before assessing their similarities. As for Jaccard, we noticed important decrease in F1-Measure; this is due to the fact that Jaccard depends on commonalities which are generally modified after enrichment. Results using KullbackLeibler showed similar behavior to other variants, except for the case that used nam as the semantic similarity measure.

In experiments using nam, all variants demonstrated peaks and irregular decrease in the curves. This is due to the particular range of values that the measure nam returns and also to the relatively slight differences among similarities of different pairs of concepts.

Finally, we report that *zhong* that has the maximum correlation coefficient with expert ratings showed the minimum decrease in F1-Measure as compared with the four other semantic similarity measures.

In all experiments, enriching representation with from the five to the seven most similar concepts, the effectiveness of all classifiers deteriorates significantly. This is due to the fact that Rocchio is dependent on text statistics and that applying Semantic Kernels introduced noise to the representation model. This had deteriorating effect on classification results according to our previous observations. Moreover, adding more concepts to the model increased in some cases the MacroAveraged F1-Measure. Taking a close look at class level, classifier in such cases declined one, two and sometimes four classes in favor of the rest of classes; this justifies the increase at Macro level.

This section presented results of experiments applying *Semantic Kernels* on five Rocchio variants using five different similarity measures *cdist*, *lch*, *nam*, *wup* and *zhong* using SNOMED-CT as a semantic resource.

To conclude, results showed significant deterioration in classification effectiveness after applying *Semantic Kernel*, this means that this approach is not beneficial to Rocchio in classifying Ohsumed documents whereas it was reported quite useful using SVM (Wang *et al.*, 2008). This is quite similar to the conclusion of authors in (Bloehdorn *et al.*, 2006) when applying Adaboost to Ohsumed corpus after enriching text representation through generalization. Enriching domain specific text representation with related concepts needs much more investigation which leads us to next experiments using another approach for enriching text representation.

4 Experiments applying scenario3 on Ohsumed using Rocchio

In these experiments we intend to enrich text representation after indexing using *Enriching Vectors* in order to assess the impact of this semantic enrichment on text classification applied in the medical domain. *Enriching Vectors* was applied to K-means for clustering and to KNN for classification (L. Huang *et al.*, 2012). To the best of our knowledge, this enrichment was not tested with any other classification technique. In this section, the platform implements Rocchio classification technique in order to evaluate its performance after applying *Enriching Vectors*. This platform implements the third scenario as described in previous chapter.

This section presents the platform for our experiments in some details and then presents the results from different points of views and concludes with some recommendation on the use *Enriching Vectors* for text classification using Rocchio.

4.1 Platform for supervised text classification deploying Enriching Vectors

In order to assess the effect of *Enriching Vectors* (scenario 3) on the process of text classification using Rocchio, we use the experimental platform illustrated in Figure 62. Similar to the previous platform, this platform uses Rocchio for training and prediction as the classification technique. As for conceptualization, same configurations are used in this platform. As for *Enriching Vectors* step, the test document vector is compared to each of the centroids learned during training. Before applying one of the classical similarity measures, the vector of the document and the vector of the centroid are mutually enriched using the proximity matrix of one of the five semantic similarity measures. After this enrichment, vectors are less sparse and share more common features (concepts). Finally, prediction step applies one of the classical similarity measures of the VSM and evaluate the results.

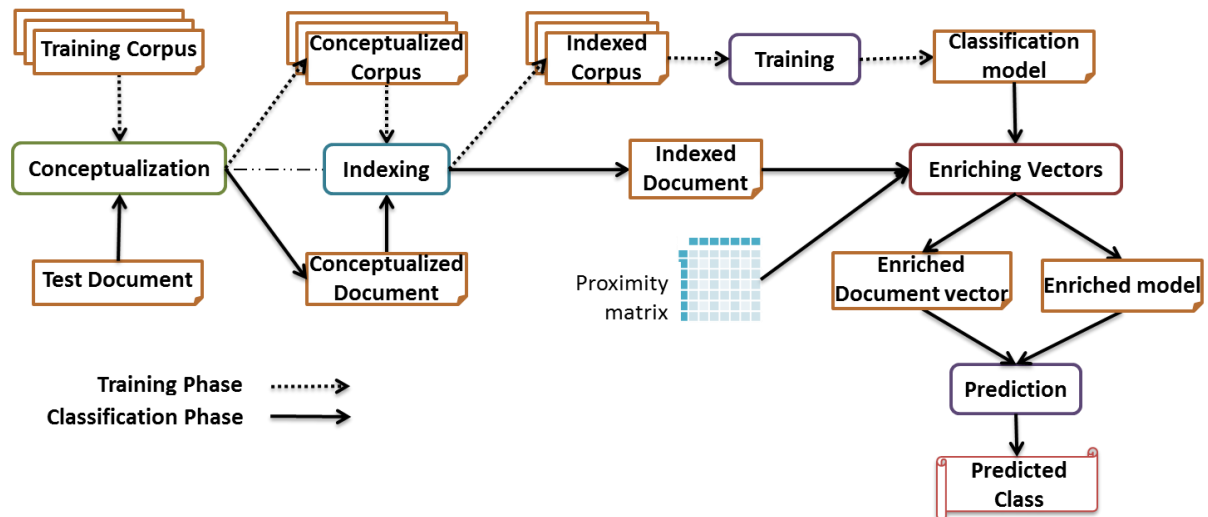


Figure 62. Platform for supervised text classification deploying Enriching vectors

Next section presents *Enriching Vectors* step in some details.

4.1.1 Enriching Vectors

Having a document d and a centroid C , for each exclusive feature c in the document its weight in the centroid is estimated using the following formula:

$$w(c, C) = w(SC(c, C)) * sim(c, SC(c, C)) * CC(c, C) \quad (78)$$

And for each exclusive feature c in the centroid, its weight in the document is estimated using the following formula:

$$w(c, d) = w(SC(c, d)) * sim(c, SC(c, d)) * CC(c, d) \quad (79)$$

Where:

$w(SC(c, V))$ is the weight of the Strongest Connection (SC) of the concept c in the vector V which is the weight of the most similar concept in V

$sim(c, SC(c, V))$ is the similarity between the concept and its strongest connection

$CC(c, V)$ is the Context Centrality (CC) of the concept c in the vector V and is given by the following formula:

$$CC(c, V) = \frac{\sum_{c_i \in V} sim(c, c_i) * w(c_i, V)}{\sum_{c_i \in V} w(c_i, V)} \quad (80)$$

Where:

$sim(c, c_i)$ is the similarity between the concept c and the concept c_i from the vector V .

$w(c_i, V)$ is the weight of concept c_i in the vector V .

4.2 Evaluating results

In these experiments, the platform executes learning five times once for each of the proximity matrices. This means that the Rocchio learns 5 different classification models or ensembles of centroids. As for classification, Rocchio uses each of the preceding models with each of its variants (Cosine, Jaccard, KullbackLeibler, Levenshtein, Pearson) resulting in $5 * 5 = 25$ executions. The detailed results from these executions that are related to each similarity measure are grouped together to analyze the impact of *Enriching Vectors* on the effectiveness of the five variants of Rocchio.

4.2.1 Results using Rocchio with Cosine

4.2.1.1 Observations

According to results illustrated in Table 39, the F1-measure obtained from applying Rocchio with Cosine similarity measure on the completely conceptualized Ohsumed corpus using concept Ids varied from (53.96%) to (72.88%) for classes (C23, C14) respectively. We report improvements in classification using Cosine after applying *Enriching Vectors* using three similarity measures cdist, nam and zhong, this increased the Macro F1-measure. Note that the

best improvement was obtained using cdist and zhong and that the improved classes in all cases were (C04) and (C23).

Enriching vectors by means of cdist semantic similarity improved the performance of Rocchio with Cosine similarity measure by a percentage that varies from (1.27%) for the class (C06) to (4.10%) for the class (C23). The absolute value of F1-measure varied from (55.37%) to (73.62%) for classes (C06, C04) respectively. Using the semantic similarity nam increased the F1-measure by (0.24%) and (2.63%) which resulted in the values (72.82%, 55.37%) for (C04) and (C23) respectively. Finally, using zhong in enriching vectors improved F1-Measure of the class (C04) with a percentage of (1.41%) and (C06) with (1.10%) and C23 with (2.41%) resulting in (73.67%, 55.28% and 55.26%) respectively.

Category/ Configuration	C04		C06		C14		C20		C23		Macro		Micro	
Original	72.65		54.68		72.88		65.20		53.96		63.87		64.81	
cdist	73.62	+1.34*	55.37	+1.27*	71.87	-1.38*	64.62	-0.89	56.17	+4.10*	64.33	+0.72	64.91	+0.15
lch	1.41	-98.06*	22.02	-59.72*	2.28	-96.87*	26.36	-59.57*	29.84	-44.69*	16.38	-74.35*	19.92	-69.26
nam	72.82	+0.24*	54.05	-1.14	72.66	-0.30*	64.90	-0.46	55.37	+2.63*	63.96	+0.14	64.69	-0.19
wup	50.75	-30.14*	41.76	-23.62	54.57	-25.13*	56.08	-13.99	50.46	-6.48*	50.72	-20.59*	50.28	-22.41
zhong	73.67	+1.41	55.28	+1.10	72.73	-0.20*	64.72	-0.74	55.26	+2.41*	64.33	+0.72	65.13	+0.50

Table 39. Results of applying Rocchio with Cosine similarity measure to Ohsumed corpus and to the results of its complete conceptualization with Enriching vectors. (*) denotes significance according to McNemar test. Values in the table are percentages.

4.2.1.2 Analysis

From previous observations we conclude that maximum increase in F1-Measure (4.10%) was obtained for the class (C23) using cdist for *Enriching Vectors*. The main particularity of this measure is that it returns values ranging between 0 and 1 and with relatively high variations between similarities of different pairs. In fact, Rocchio with Cosine obtained the lowest F1-Measure (53.96%) on this class using the conceptualized corpus.

Previous reported improvements at class level influenced the MacroAveraged F1-Measure with a gain of (0.72%, 0.14% and 1.08%) using strategies cdist, nam and zhong respectively. Note that we have no evidence that the overall performance of Rocchio using Cosine on the conceptualized corpus is significantly different from its performance on the corpus after applying *Enriching Vectors* using either semantic similarity measure according to McNemar test.

In fact, enriching text representation using similar concepts is beneficial to classifying three classes of documents (C04, C06, C23) with either cdist or zhong. Moreover, this enrichment is useful to classifying classes (C04 and C23) with nam semantic similarity measure.

According to Figure 63, using zhong or cdist increased F1-Measure of three classes which improved the overall performance of Rocchio with Cosine. This improvement is higher than the one reported with nam that results in a MacroAveraged F1-measure of (64.33%) as

presented formerly in Table 39. Note that both measures return low values in the range [0,1] so using them to modify weights of features in BOCs wouldn't affect the weighting scheme.

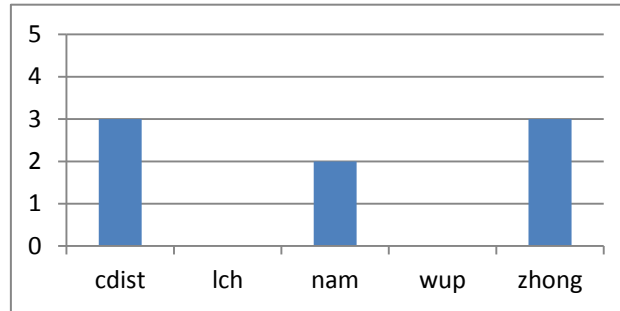


Figure 63. Number of improved classes after applying Enriching Vectors on Rocchio with Cosine using five semantic similarity measures

4.2.2 Results using Rocchio with Jaccard

4.2.2.1 Observations

Results of applying Rocchio with Jaccard on conceptualized Ohsumed after text representation enrichment are illustrated in Table 40. The F1-measure obtained from test on the corpus before enrichment varied from (47.40%) to (73.29%) for classes (C23, C14) respectively. We report improvements in classification using Jaccard after applying *Enriching Vectors* using three similarity measures cdist, nam and zhong, this increased the Macro F1-measure. Note that the best improvement obtained was using cdist and that the improved classes in all cases were (C04) and (C23).

Category/ Configuration	C04		C06		C14		C20		C23		Macro		Micro	
Original	72.76		53.45		73.29		65.39		47.40		62.46		63.92	
cdist	73.58	+1.12	53.73	+0.52	73.54	+0.34*	65.69	+0.46	51.88	+9.45*	63.68	+1.96*	64.99	+1.66
lch	0.16	-99.78*	0.58	-98.91*	0.90	-98.78*	22.69	-65.30*	-	-	6.08	-90.26*	12.70	-80.13
nam	73.02	+0.35	52.76	-1.30	73.25	-0.05*	65.27	-0.17	49.16	+3.71*	62.69	+0.37	64.07	+0.22
wup	43.20	-40.63*	34.82	-34.86*	48.55	-33.76*	33.80	-48.31*	16.61	-64.97*	35.39	-43.33*	36.00	-43.69
zhong	72.93	+0.23	53.07	-0.71	73.29	+0.01	65.50	+0.16	48.73	+2.79*	62.70	+0.39	64.25	+0.50

Table 40. Results of applying Rocchio with Jaccard similarity measure to Ohsumed corpus and to the results of its complete conceptualization with Enriching vectors. (*) denotes significance according to McNemar test. Values in the table are percentages.

Using cdist for enriching vectors improved the performance of Rocchio with Jaccard similarity measure by a percentage that varies from (0.34%) for the class (C14) to (9.45%) for the class (C23). The absolute value of F1-measure varied from (51.88%) to (73.58%) for classes (C23, C04) respectively. Using the semantic similarity nam increased the F1-measure by (0.35%) and (3.71%) which resulted in the values (73.02%, 49.16%) for (C04) and (C23) respectively. Finally, using zhong in enriching vectors improved F1-Measure of the class (C04)

with a percentage of (0.23%), (C14) with (0.01%), (C20) with (0.16%) and C23 with (2.79%) resulting in (72.93%, 73.29%, 65.50%, and 48.73%) respectively.

4.2.2.2 Analysis

The maximum increase in F1-Measure (9.45%) was obtained for the class (C23) using *cdist* for *Enriching Vectors*. The main particularity of this measure is the range $[0,1]$ and the variations of values it returns. In fact, Rocchio with Jaccard obtained the lowest F1-Measure (47.40%) using the conceptualized corpus for this particular class.

Previous reported improvements at class level influenced the MacroAveraged F1-Measure with a gain of (1.96%, 0.37% and 0.39%) using strategies *cdist*, *nam* and *zhong* respectively. Note that the overall performance of Rocchio using Jaccard on the conceptualized corpus is significantly different from its performance on the corpus after applying *Enriching Vectors* using *cdist* semantic similarity measure according to McNemar test.

In fact, enriching text representation using similar concepts is beneficial to classifying five, two and four classes with *cdist*, *nam* and *zhong* respectively (see Figure 64). Moreover, the system showed better classification results on (C04 and C23) for all of the preceding similarities. In all of these cases, the increase in F1-Measure increased the MacroAveraged F1-Measure. Finally, best results are obtained by applying *Enriching Vectors* using Jaccard and *cdist* as a semantic similarity that resulted in a MacroAveraged F1-Measure of (63.68%) (see Table 40). Note that *cdist* returns low values in the range $[0,1]$ so using them to modify weights of features in BOCs wouldn't affect the weighting scheme.

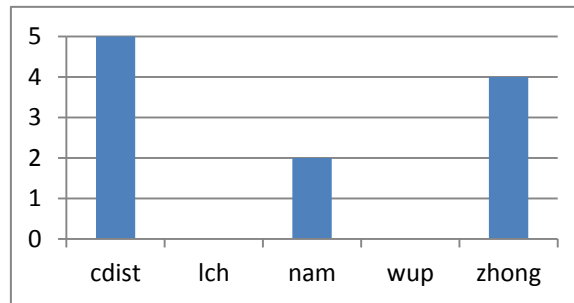


Figure 64. Number of improved classes after applying Enriching Vectors on Rocchio with Jaccard using five semantic similarity measures

4.2.3 Results using Rocchio with Kulback

Detailed results of applying *Enriching Vectors* on text representation and then testing Rocchio with KullbackLeibler on the resulting vectors are in Table 41. We report deterioration in the performance of Rocchio after vector enrichment. The particularity of KullbackLeibler as compared with other similarity measures is related to the fact that it considers the divergence between feature distributions among documents. Obviously, these distributions change after enrichment which complicates the prediction process.

Category/ Configuration	C04		C06		C14		C20		C23		Macro		Micro	
Original	71.11		68.39		77.78		64.68		57.69		67.93		68.28	
cdist	9.68	-86.38	22.89	-66.53	6.52	-91.61	8.38	-87.05	26.04	-54.86	14.70	-78.36	18.26	-73.26
lch	-	-	21.53	-68.52	-	-	0.29	-99.55	23.71	-58.89	15.18	-77.66	15.27	-77.64
nam	17.54	-75.34	30.16	-55.89	14.70	-81.11	36.10	-44.18	35.46	-38.52	26.79	-60.56	28.15	-58.77
wup	30.11	-57.65	18.92	-72.33	1.26	-98.38	0.59	-99.08	38.95	-32.48	17.97	-73.55	27.57	-59.62
zhong	0.63	-99.12	21.94	-67.91	-	-	-	-	2.58	-95.52	8.39	-87.66	12.56	-81.60

Table 41. Results of applying Rocchio with KullbackLeibler similarity measure to Ohsumed corpus and to the results of its complete conceptualization with Enriching vectors. (*) denotes significance according to McNemar test. Values in the table are percentages.

4.2.4 Results using Rocchio with Levenshtein

In these experiments (see Table 42), we observed improvements in two cases only: class (C23) using nam with a percentage of (0.69%) and class (C20) using zhong with a percentage of (6.56%). These improvements resulted in (41.32%) and in (58.41%) correspondingly. These improvements were limited at class level and had no effect at the Macro level. The deterioration in Rocchio's effectiveness after applying *Enriching Vectors* on text representation is related to the fact that Levenshtein is based on the difference between the compared vectors. This difference seems to be affected after enrichment as the compared vectors become less sparse.

Category/ Configuration	C04		C06		C14		C20		C23		Macro		Micro	
Original	72.83		50.41		68.44		54.82		41.03		57.51		58.87	
cdist	45.50	-37.52	42.91	-14.88	58.48	-14.56	46.26	-15.60	39.69	-3.28	46.57	-19.02	17.26	-70.68
lch	0.00	-	0.00	-	0.00	-	0.32	-99.42	23.50	-42.74	4.76	-91.72	20.70	-64.83
nam	41.55	-42.95	42.24	-16.20	57.56	-15.91	45.73	-16.58	41.32	+0.69	45.68	-20.57	26.69	-54.66
wup	23.95	-67.12	24.55	-51.30	2.36	-96.55	2.72	-95.04	27.15	-33.83	16.15	-71.92	18.82	-68.02
zhong	42.74	-41.31	30.39	-39.71	50.35	-26.43	58.41	+6.56*	33.89	-17.41	43.16	-24.95	23.32	-60.39

Table 42. Results of applying Rocchio with Levenshtein similarity measure to Ohsumed corpus and to the results of its complete conceptualization with Enriching vectors. (*) denotes significance according to McNemar test. Values in the table are percentages.

4.2.5 Results using Rocchio with Pearson

4.2.5.1 Observations

Using Rocchio and Pearson for text classification after applying *Enriching Vectors* resulted in some improvements at the Macro level. The F1-measure obtained from test on the corpus before enrichment varied from (54.20%) to (72.38%) for classes (C23, C04) respectively. Only two similarity measures cdist and zhong increased Rocchio's Macro F1-measure. Note that the best improvement obtained was using zhong on the class (C23). Detailed results are presented in Table 43.

Category/ Configuration	C04		C06		C14		C20		C23		Macro		Micro	
Original	72.38		54.34		72.22		64.92		54.20		63.61		64.45	
cdist	72.57	+0.25	54.61	+0.50	71.88	-0.47	64.86	-0.08	54.73	+0.97*	63.73	+0.19	64.43	-0.03
lch	0.63	-99.13*	16.56	-69.52*	8.28	-88.53*	24.27	-62.62*	0.67	-98.76*	10.08	-84.15*	15.15	-76.49
nam	71.93	-0.63*	53.39	-1.75	72.63	+0.57	64.42	-0.76*	53.30	-1.65*	63.13	-0.75	64.02	-0.65
wup	58.69	-18.92*	42.35	-22.05*	68.08	-5.72*	51.05	-21.37*	33.47	-38.25*	50.73	-20.25*	51.54	-20.02
zhong	72.58	+0.27	54.36	+0.05*	72.50	+0.39*	65.27	+0.55*	55.73	+2.82*	64.09	+0.75	64.85	+0.62

Table 43. Results of applying Rocchio with Pearson similarity measure to Ohsumed corpus and to the results of its complete conceptualization with Enriching vectors. (*) denotes significance according to McNemar test. Values in the table are percentages.

Using cdist for enriching vectors improved the performance of Rocchio with Pearson similarity measure of three classes (C04, C06, C23) by the percentages (0.25%, 0.50%, 0.97%) resulting in the absolute values of F1-measure (72.57%, 54.61%, 54.73%) respectively. Using the semantic similarity nam increased the F1-measure by (0.57%), this resulted in (72.63%) of F1-Measure for (C14). Finally, using zhong in enriching vectors improved F1-Measure of the five classes with a percentage that varied between (0.05%) and (2.82%) for (C06) and (C23) respectively. Resulting F1-Measure values varied between [54.36%, 72.58%].

4.2.5.2 Analysis

According to observations on the detailed results, we report a maximum increase in F1-Measure of (2.82%) for the class (C23) using zhong for Enriching Vectors. In fact, Rocchio with Pearson obtained the lowest F1-Measure (54.20%) using the conceptualized corpus for this particular class.

Previous reported improvements at class level influenced the MacroAveraged F1-Measure with a gain of (0.19% and 0.75%) using strategies cdist and zhong respectively. We have no evidence that the overall performance of Rocchio using Pearson on the conceptualized corpus is significantly different from its performance on the corpus after applying Enriching Vectors using either semantic similarity measure according to McNemar test.

In fact, enriching text representation using similar concepts is beneficial to classifying three, one and five classes with cdist, nam and zhong respectively (see Figure 65). Moreover, the system showed better classification results on (C04, C06, and C23) using cdist and zhong similarities. In both cases, the increase in F1-Measure increased the MacroAveraged F1-Measure. Rocchio with Pearson gave best results by applying Enriching Vectors using zhong as a semantic similarity measure; this resulted in a MacroAveraged F1-Measure of (64.09%) (see Table 43). Note that zhong returns low values in the range [0, 1].

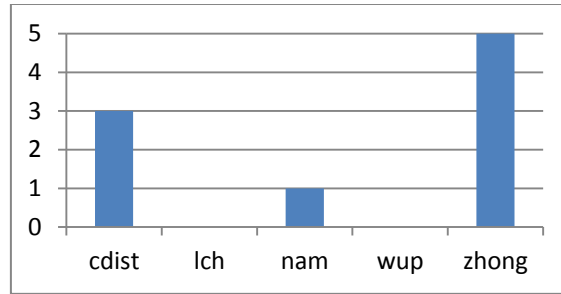


Figure 65. Number of improved classes after applying Enriching Vectors on Rocchio with Pearson using five semantic similarity measures

4.2.6 Conclusion

Previous sections presented an experimental study on the effects of *Enriching Vectors* on Rocchio's performance using five semantic similarity measures (cdist, lch, nam, wup, zhong) on concepts of SNOMED-CT pair-to-pair. Tests were realized on completely conceptualized Ohsumed using Ids of the best mappings. As for prediction, we used five classical similarity measures: Cosine, Jaccard, KullbackLeibler, Levenshtein, Pearson.

After a detailed presentation of observed results, here we summarize with some important points. First of all, in all cases, using the semantic similarities lch and wup caused deterioration in Rocchio's performance while other similarities showed some improvements. Note that the only aspect that cdist, nam, and zhong share is the relatively low values of semantic similarity they return as compared to both lch and wup which justifies their different influence of text representation. Best overall performance was obtained using Rocchio with Cosine and zhong similarity measure with a MacroAveraged F1-Measure of (64.33%). This value is higher than the one reported in (L. Huang *et al.*, 2012) where authors tested *Enriching Vectors* on a small corpora retrieved from Ohsumed using KNN classifier.

Second, we distinguish two groups of Rocchio variants according to their performance after applying *Enriching Vectors*: first group contains Cosine, Jaccard and Pearson and the second one contains KullbackLeibler and Levenshtein. The main difference between these groups is that the first one assesses similarity among vectors using their commonalities whereas the second one depends on their differences in order to assess their similarities. In general, *Enriching Vectors* aims to reduce the sparseness of text representation; this seems to help the first group in assessing similarities. On the contrary, this enrichment seems to be harmful to assessing similarities as the differences between vectors are modified after enrichment.

Third, when the system performance using a specific method has a low F1-measure value, as it the case for the class (C23), *Enriching Vectors* can improve this value with a maximum gain reaching (9.45%) in the case of Rocchio with Jaccard. Similar to our observations after applying conceptualization, the class "C23" is very large compared to others and so enriching class representation with similar concepts might result in a better identification of this class which led to better results.

Finally, it seems beneficial to Rocchio-based classification to apply *Enriching Vectors* before prediction as it modifies the behavior of the classifier and can improve its effectiveness. This is true as compared to the baseline that used Rocchio with Cosine on the conceptualized

corpus using (Complete+Best+Id). However, resulting performance is dependent on the semantic similarity measure used in enrichment and also on the similarity measure used for prediction. Consequently, it necessary to verify experimentally and to check whether *Enriching Vectors* is useful in a particular context. Note that, Rocchio with Cosine using Add conceptualization strategy is more effective than its performance after applying *Enriching Vectors* on conceptualized corpus using (Complete+Best+Id).

So far, the exploitation of semantic resources mainly focused on the representation step of the classification process. Next section, presents and experimental study on deploying semantics during prediction step of Rocchio-based classification.

5 Experiments applying scenario4 on Ohsumed using Rocchio

In these experiments we intend to use *Semantic Text-To-Text Semantic Measures* and to assess their impact on text classification applied in the medical domain. To the best of our knowledge, the effect of such measures on the performance of Rocchio wasn't thoroughly investigated in the context of supervised text classification. In this section, the platform implements Rocchio classification technique using *Semantic Text-To-Text Semantic Measures* as the similarity measure for class prediction. This platform implements the forth scenario as described in previous chapter.

This section presents the platform for our experiments in some details and then presents the results from different points of views and concludes with some recommendation on the use *Semantic Text-To-Text Semantic Measures* for text classification using Rocchio.

5.1 Platform for supervised text classification deploying Semantic Text-To-Text Similarity Measures

In order to assess the effect of *Semantic Text-To-Text Similarity Measures* on the process of text classification using Rocchio, we use the experimental platform illustrated in Figure 66. Similar to the previous platform, this platform uses Rocchio for training and prediction as the classification technique. As for conceptualization, same configurations are used in this platform but no enrichment is applied to the indexes. As for *prediction* step, the test document vector is compared to each of the centroids learned during training. Instead of applying one of the classical similarity measures, the platform uses a *Semantic Text-To-Text Similarity Measures* to assess the similarity between the vector of the document and the vector of the centroid. These measures are aggregation functions on the semantic similarity between their concepts pair-to-pair.

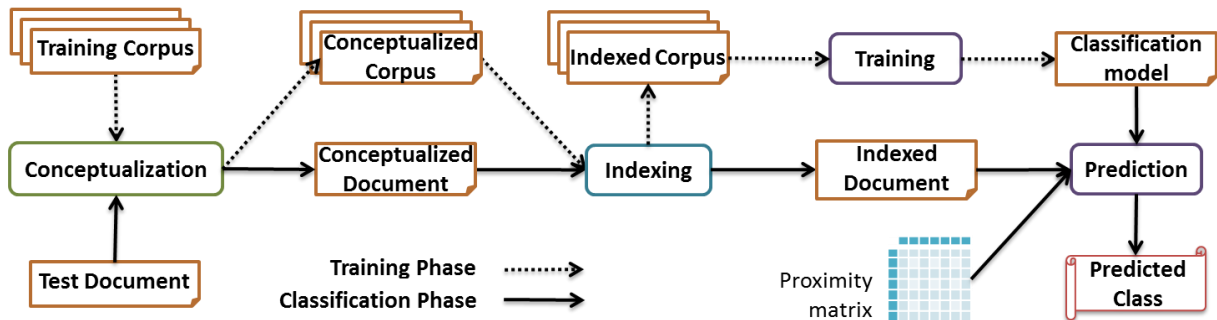


Figure 66. Platform for supervised text classification deploying Semantic Similarity Measures

Next section presents *Semantic Text-To-Text Similarity Measures* in some details.

5.1.1 Semantic Text-To-Text Similarity Measures

Previous chapter presented five different aggregation functions for assessing *Semantic Text-To-Text Similarity*. Most of these measures are based on an average of the similarities between the concepts of the compared vectors pair to pair taking into account some of the vocabulary

statistics. Our empirical study using Rocchio showed that it is very essential to use these statistics in the aggregation function in order to use the function as a similarity measure for class prediction. Consequently, we focus on the last two similarity measures presented in the previous chapter. The first measure was proposed in (Mihalcea *et al.*, 2006). This measure, called by *AvgMaxAssymIdf*, is based on the pairs of concepts having maximal similarities among the compared vectors and the corresponding Inverse Document Frequency (IDF) of them according to the following formula:

$$SemSim(d, C) = \frac{1}{2} \left(\frac{\sum_{c \in d} maxSim(c, C) * idf(c)}{\sum_{c \in d} idf(c)} + \frac{\sum_{c \in C} maxSim(c, d) * idf(c)}{\sum_{c \in C} idf(c)} \right) \quad (81)$$

Where:

$maxSim(c, C)$ is the maximum similarity between the concept (c) and all concepts in the centroid (C)

$idf(c)$ is the inverse document frequency of the concept (c)

We proposed in previous chapter a new aggregation function *AvgMaxAssymTFIDF* adapting the previous one to text classification by using TFIDF weights instead of IDF weights in order to take into account the significance of a concept in a document instead of its significance in the corpus. This function becomes as the following formula:

$$SemSim(d, C) = \frac{1}{2} \left(\frac{\sum_{c \in d} maxSim(c, C) * Tfidf(c)}{\sum_{c \in d} Tfidf(c)} + \frac{\sum_{c \in C} maxSim(c, d) * Tfidf(c)}{\sum_{c \in C} Tfidf(c)} \right) \quad (82)$$

Where:

$maxSim(c, C)$ is the maximum similarity between concept (c) and all concepts in (C)

$Tfidf(c)$ is the normalized frequency of the concept (c) according to the TFIDF weighting scheme.

5.2 Evaluating results

In these experiments, the platform executes classification five times once for each of the proximity matrices and once for each aggregation function. This means that the Rocchio learns once a unique classification model. As for classification, Rocchio uses each of the aggregation function in prediction using one of the five proximity matrices resulting in $5 * 2 = 10$ executions. The detailed results from these executions that are related to each semantic similarity measure (between concepts pair-to-pair) are grouped together to analyze the impact of *Semantic Text-To-Text Similarity measures* on the effectiveness of Rocchio. In next subsections, we use as a baseline of comparison Rocchio with Cosine applied on conceptualized Ohsumed using the strategy (Complete+Best+Ids).

5.2.1 Results using AvgMaxAssymIdf

Results of these experiments are detailed in Table 44. We notice that using *AvgMaxAssymIdf* semantic similarity measure for prediction in Rocchio didn't improve its performance at MacroAveraged level. Nevertheless, local significant improvements occurred when treating documents related to (C06) that is one of the least populated classes in the training corpus. This improvement varied from (5.44%) using wup to (15.16%) using lch resulting in F1-Measure

ranging between (57.65%) and (62.97%). These improvements are statistically significant according to McNemar test. Other improvements occurred as well; the first is significant using lch on (C04) and the second using cdist on (C14). Note that the class (C06) is the least populated class among the five treated classes.

Category/ Configuration	C04		C06		C14		C20		C23		Macro		Micro	
Original	72.65		54.68		72.88		65.20		53.96		63.87		64.81	
cdist	71.90	-1.03	62.56	+14.41*	73.46	+0.80	56.74	-12.97*	35.07	-35.01*	59.95	-6.15*	62.74	-3.19
lch	73.41	+1.05*	62.97	+15.16*	71.52	-1.87*	57.89	-11.21*	23.82	-55.85*	57.92	-9.32*	62.06	-4.24
nam	71.40	-1.72	58.74	+7.43*	71.69	-1.62	51.94	-20.34*	29.98	-44.44*	56.75	-11.15*	60.59	-6.50
wup	68.12	-6.23*	57.65	+5.44*	69.26	-4.97*	44.47	-31.80*	21.20	-60.71*	52.14	-18.37*	57.70	-10.96
zhong	71.96	-0.95	60.49	+10.63*	72.49	-0.53	54.88	-15.82*	36.64	-32.10*	59.29	-7.17*	62.04	-4.27

Table 44. Results of applying Rocchio with AvgMaxAssymIdf semantic similarity measure to Ohsumed corpus and to the results of its complete conceptualization. (*) denotes significance according to McNemar test. Values in the table are percentages.

5.2.2 Results using AvgMaxAssymTFIDF

5.2.2.1 Observations

Using AvgMaxAssymTFIDF as the similarity measure for Rocchio prediction improved the classification of (C06). This improvement is high using all of the five semantic similarity measures ranging between (16.46%) and (18.13%) for nam and wup respectively. These improvements led to a better F1-Measure in the range [63.68%, 64.60%] as compared with results using Cosine as similarity measure on the same class (54.68%). Detailed results are in Table 45.

Category/ Configuration	C04		C06		C14		C20		C23		Macro		Micro	
Original	72.65		54.68		72.88		65.20		53.96		63.87		64.81	
cdist	74.75	+2.89*	64.56	+18.07*	75.55	+3.67*	59.31	-9.03*	52.45	-2.79*	65.32	+2.27*	66.91	+3.25
lch	76.25	+4.96*	64.39	+17.76*	73.74	+1.19*	56.45	-13.43*	49.17	-8.88*	64.00	+0.20	66.27	+2.26
nam	74.80	+2.96*	63.68	+16.46*	71.60	-1.76*	57.22	-12.23*	44.29	-17.91*	62.32	-2.43	64.57	-0.37
wup	74.79	+2.94*	64.60	+18.13*	73.01	+0.18	50.92	-21.89*	40.79	-24.41*	60.82	-4.78	64.59	-0.34
zhong	74.65	+2.75*	64.23	+17.47*	75.26	+3.26*	59.74	-8.38*	50.55	-6.31*	64.89	+1.59*	66.57	+2.72

Table 45. Results of applying Rocchio with AvgMaxAssymTFIDF semantic similarity measure to Ohsumed corpus and to the results of its complete conceptualization. (*) denotes significance according to McNemar test. Values in the table are percentages.

Using all measures, except for nam, improved the F1-Measure of classes (C04) and (C14), these improvements are lower if compared to those on (C06). As for (C04), the improvements ranged from (2.75%) to (4.96%) using zhong and lch respectively resulting in F1-Measure in [74.65%, 76.25%]. On the other hand, improvements treating (C14) ranged from

(0.18%) to (3.67%) using wup and cdist respectively resulting in F1-Measure in [73.01%, 75.55%]. Only three similarity measures cdist, lch and zhong increased Rocchio's Macro F1-measure.

5.2.2.2 Analysis

Previous observations showed that the maximum increase in F1-Measure occurred when treating the class (C06) and is of a percentage of (18.13%) using lch for *Semantic Text-To-Test Similarity measure*. In fact, this class is the least populated class in the corpus and Rocchio with Cosine obtained on the completely conceptualized corpus a relatively low value of F1-Measure for this class.

These improvements at class level influenced the MacroAveraged F1-Measure with a gain ranging from (0.20%) to (2.27%) using semantic similarities lch and cdist respectively. In fact, the overall performance of Rocchio using Cosine on the conceptualized corpus is significantly different from its performance on the corpus after applying *AvgMaxAssymTFIDF* according to McNemar test and using two semantic similarity measures zhong and dist.

In fact, enriching text representation using similar concepts is useful to classifying three classes for all semantic similarities except for nam that helped Rocchio improve its performance on two classes only (see Figure 67). Using cdist, lch or zhong, the increase in F1-Measure at class level increased the MacroAveraged F1-Measure. This approach has no impact on the weighting scheme which makes it less sensitive than others of different ranges of values returned by these measures. Rocchio with *AvgMaxAssymTFIDF* gave best results by using cdist as a semantic similarity measure; this resulted in a MacroAveraged F1-Measure of (65.32%) (see Table 45). Note that cdist returns low values in the range [0, 1].

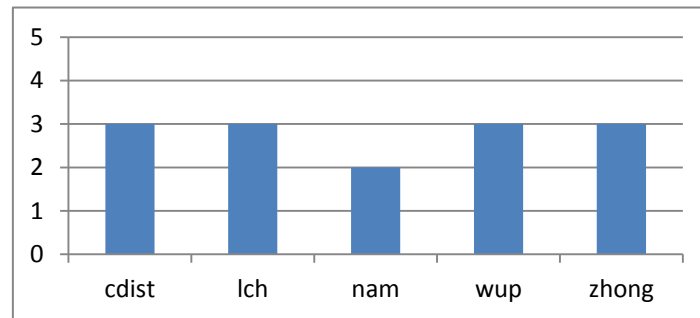


Figure 67. Number of improved classes after applying Rocchio with *AvgMaxAssymTFIDF* for prediction

5.2.3 Conclusion

Previous sections presented an experimental study on the effects of *Semantic Text-To-Test Similarity Measures* on Rocchio's prediction using two different aggregation functions *AvgMaxAssymIdf* and *AvgMaxAssymTFIDF*. These functions used five semantic similarity measures (cdist, lch, nam, wup, zhong) on concepts of SNOMED-CT pair-to-pair. Tests were realized on completely conceptualized Ohsumed using Ids of the best mappings.

To sum up, here we list our conclusions on the experimental study using semantic similarity measures with Rocchio for prediction. First of all, all semantic similarity measures

improved Rocchio's performance for the class C06. Nevertheless, only three cases using *AvgMaxAssymTFIDF* improved results at MacroAveraged level. Best overall performance occurred with Rocchio and cdist similarity measure with a MacroAveraged F1-Measure of (65.32%). Both similarity measures: wup and lch, improved the performance of Rocchio at class level.

Second, we distinguish two important points for developing *Semantic Text-To-Text Similarity Measures*. The first point is that these measures worked with the five similarity measures and especially with cdist, lch and zhong. This means that they are less sensitive to differences between the ranges of the values returned by these measures, which was not the case with *Enriching Vectors*. The second point is related to the aggregation functions themselves; the function *AvgMaxAssymTFIDF* showed much better results than *AvgMaxAssymIdf* as it takes into account the TFIDF weighting model in assessing similarities between a document and a centroid. In fact, it is essential to an aggregation function to take into account language and text statistics in assessing similarities.

Third, least populated classes like (C06) are challenging for classification technique as compared to other classes for which the classification model is much easier to learn. However, *Semantic Text-To-Text Similarity Measures* helped the classifier distinguish this class with a maximum gain reaching (18.13%) in the case of *AvgMaxAssymTFIDF* using lch. Similar to our observations after applying conceptualization, the class "C06" is among the least populated classes as compared to others and so using *Semantic Text-To-Text Similarity Measures* might result in a better identification of this class which led to better results.

Finally, it seems beneficial to Rocchio-based classification to apply *Semantic Text-To-Text Similarity Measures* for prediction as it modifies the behavior of the classifier and can improve its effectiveness. However, resulting performance is dependent on the semantic similarity measure and the aggregation function used in prediction. Consequently, it is necessary to develop *Semantic Text-To-Text Similarity Measures* that are adapted to the application context.

6 Conclusion

This chapter presented an experimental study to investigate the influence of semantics on the classification process. We involved concepts before indexing through *Conceptualization*, then concepts and relation after indexing through enrichment using either *Semantic Kernels* or *Enriching Vectors*. Moreover, we involved semantics in prediction through *Semantic Text-To-Text Similarity*. As this work main interest is in the medical domain, we used the Ohsumed corpus and the UMLS as the semantic resource for conceptualization that we tested on Rocchio's variants, SVM and NB, and then SNOMED-CT for the rest of the study that was applied to Rocchio's variants only for reasons related to efficiency and extendibility.

Concerning *Conceptualization* (scenario1), different conceptualization strategies were tested in this work this led to different results according to the classification technique. In general, all techniques preferred adding semantics into text rather than using them exclusively or substitute words by concepts. Even if concepts express explicit semantics and help the classical representation model BOW overcome its limit, yet words are still needed in the process of classification and the best alternative feature to words is words and concepts together in the feature space. Moreover, techniques that are highly dependent on the representation model as Rocchio prefer integrating concepts in text and treat them like other phrases or words in text. On the other hand SVM and NB used entire concepts in indexing which improved their performance significantly.

Concerning *Semantic Kernels* (scenario2), results showed deterioration in the performance of Rocchio and its variants after applying *Semantic Kernels* on vectors that represent corpus documents. This applies to all the used semantic similarities and any number of similar concepts used in enrichment. The conclusion of authors in (Bloehdorn *et al.*, 2006) confirms this matter. Thus, *Semantic Kernels* introduce noise to the text representation and weakens its capability to distinguish classes.

Concerning *Enriching Vectors* (scenario3) returned better results as compared to *Semantic Kernels*. Nevertheless, this improvement depends on the semantic similarity measure used in enrichment and particularly on the range of values it returns. Moreover it depends on the similarity measure used in prediction, as only three variants of Rocchio out of five showed improved results after enrichment.

Concerning *Semantic Text-To-Text Similarity Measures* (scenario4) that were used in prediction instead of the classical similarity measures of the VSM, test showed better results than the preceding enrichment and especially when taking the weighting scheme into account in the aggregation function. Thus, it seems beneficial to Rocchio-based classification to apply *Semantic Text-To-Text Similarity Measures* for prediction as it modifies the behavior of the classifier and can improve its effectiveness.

The main difference between the last two approaches is that *Semantic Text-To-Text Similarity Measures* uses semantic similarities and text statistics in prediction, whereas in *Enriching Vectors*, the semantic similarity measures are used to modify text representation and the centroids as well. This justifies that *Semantic Text-To-Text Similarity Measures* are less sensitive than *Enriching Vectors* to differences between the five used semantic similarity

measures. This includes differences between the ranges of values they return and between their basic principles.

Finally, involving semantics in text classification process does modify its behavior. Maximum improvements were obtained after involving semantics in text representation through *Conceptualization*. Moreover, all approaches (with few exceptions) showed improvements in effectiveness when classifying three classes (C23, C06, and C20). This confirms our assumptions in chapter 2 on the use of semantics and its influence in cases of large classes (C23) and low populated classes (C06 and C20). Although this approach did not improve the performance of classification sufficiently, Semantic similarity measures and aggregation function seem to be more adequate than classical similarity measures to compare vectors of concepts rather than classical similarity measures of the VSM.

CHAPTER 6: CONCLUSION AND PERSPECTIVES

1 Conclusion

Organizing objects into classes dates back to Plato while classifying text documents dates back to at least 26 B.C.E. This classification has been always a great interest to humans aiming to discover the thematic relations among texts, which facilitates access to large databases and relieves humans from memorizing large amounts of information. The classification of natural language texts is a challenging task to automate because it requires expertise that only domain experts have. For example, librarians learn how to assign metadata to documents to describe their subjects according to a particular classification system. This kind of systems involves years of work to develop and a great labor to learn and to utilize.

Supervised text classification is an automated way to thematically classify text documents in a predetermined list of classes or categories. The main challenge for such techniques is to make computers learn the expertise needed for classification in order to be able to classify any new document introduced into its system. This means that computers must have similar ways of text perception and interpretation as humans.

Chapter 2 of this thesis presented some details on supervised text classification: its origins, history and commonly used classical supervised techniques: Rocchio, SVM and NB. Traditionally, most text classification techniques use BOW for text representation that has three shortcomings: it ignores synonymy, ambiguity and relations between features of the vector space. Chapter 2 presents an experimental study applying three traditional classification techniques on three different corpora. Results showed that the performance of a classification technique depends on the context of the task; no classification technique is the best for all tasks. Moreover, Rocchio showed some difficulties in classification that led to investigate the effect of corpus labeling on its effectiveness. In fact, Rocchio's effectiveness is affected when dealing with similar classes, general classes and heterogeneous classes. Chapter 2 proposed to overcome these limitations by means of semantic resources; redefining centroids in the concept space might limit intersections between spheres of similar classes.

In fact, involving semantics in the process classification aims to make computer perception of natural language text meet or even get closer to humans' perception and interpretation. For examples humans resolve ambiguities and can distinguish the meaning the words convey. In addition, humans do not treat words independently as a structure of words may convey meaning as well. Finally, semantics can be involved at different steps of the classification process: text representation, training and prediction.

Chapter 3 presented a review on the state of art works involving semantics in text classification and other tasks in the domain of IR. Different sources of semantics were deployed in these works: General purpose resources like WordNet and Wikipedia and domain specific ones like UMLS, MeSH and AGROVOC. Different levels for integrating semantics are possible as well starting from text representation to classification model and finally the class prediction or text to text comparison. Many of these approaches reported significant improvement in effectiveness after integrating semantics. Moreover, many authors focused on problems related to specific domains and particularly when dealing with the medical domain and argued the utility of using specific domain ontologies instead of general purpose ones in such contexts.

Most reviewed works investigated the effect of semantics on text treatment at the representation level after indexing. In general, they deployed explicit semantics as specified in ontologies through concepts generating *BOC* as a model for text representation. *Conceptualization* is the process of mapping text to concepts that we deployed to enrich the original BOW in order to overcome its limitations. Other works deployed semantic similarity between concepts of the semantic resources to enrich text representation and also to assess text-to-text similarity. As for representation enrichment, we distinguished three major approaches: *semantic kernels* usually used with SVM, *generalization* introduces noise in domain specific tasks and *Enriching Vectors* that enriches pairs of documents mutually. Considering text-to-text semantic similarity, most approaches are aggregation functions on semantic similarity between concepts pair-to-pair. These approaches were developed in an ad hoc manner and need to be tested in large scale applications.

Despite the promising results, integrating semantics in classification is a subject of debate state of the art works seem to disagree on its utility. Nevertheless, it seems to be promising to take the application domain into consideration when developing a system for semantic classification. This led us to propose generic testbeds to support semantic integration at different levels in the text classification process and to investigate its influence on text classification effectiveness in the medical domain.

Chapter 4 presented a conceptual framework for involving semantics in text classification; using concepts in conceptualization and semantic similarities in the other approaches. We proposed four scenarios to apply these approaches: *Conceptualization only*, *Conceptualization and enrichment before training*, *Conceptualization and enrichment before prediction* and *Conceptualization and semantic text-to-text similarity for prediction*. In addition, this chapter presented many tools for the medical domain that we found effective in realizing text conceptualization and in calculating semantic similarities as well. We chose to use UMLS as the semantic resource, MetaMap for text to concept mapping and Ohsumed as the text collection.

Chapter 5 presented an experimental study to investigate the influence of semantics on the classification process. We implemented the four preceding scenarios in four platforms in order to assess the influence of UMLS on classification effectiveness using Ohsumed. We tested *Conceptualization* using 12 different strategies and three different classification techniques: SVM, NB and Rocchio. Results demonstrated that involving concepts in text before indexing improves classification effectiveness. Then we tested *Conceptualization and enrichment before training* using *Semantic Kernels* to enrich text representation using concepts and relations between them after indexing. This method introduced noise into text representation and caused deterioration in Rocchio's performance. Starting from this scenario, Rocchio is the only tested techniques for reasons related to its efficiency and extendibility, UMLS is reduced to SNOMED-CT which is used as the semantic resource and we use complete conceptualization strategy or a pure BOCs for text representation. Concerning *Conceptualization and enrichment before prediction*, experimental results using Rocchio on Ohsumed showed that this mutual enrichment of two vectors enhances the effectiveness of classical similarity measures like Cosine. In fact, this approach reduces sparseness of the compared vectors of concepts and

increases the number of features they share. Finally, we tested *Conceptualization and semantic text-to-text similarity for prediction* using two different aggregation functions. Results demonstrated that these function may be more adequate to compare BOCs than classical similarity measures like Cosine.

2 Contribution

In this work, we implemented four different platforms and tested them using different classification techniques and different parts of UMLS in order to assess the impact of semantics on text classification in order to prove the statement:

Using concepts in text representation and taking the relations among them into account during the classification process can significantly improve the effectiveness of text classification, using classical classification techniques.

2.1 Text conceptualization

UMLS Concepts were involved in text classification through text *Conceptualization* (scenario1), using different conceptualization strategies. The impact of these strategies on SVM, NB and the five variants of Rocchio was not identical.

- Adding concepts into text is the preferred strategy; classification techniques preferred adding semantics into text rather than using them exclusively or substituting words by concepts. Even if concepts express explicit semantics and help the classical representation model BOW overcome its limit, yet a combination of concepts and words seems to be the best alternative to words only.
- Enriching text with concepts and apply indexing on them as the rest of text is preferred with classification techniques that are highly dependent on the representation model and text statistics as Rocchio; it prefers integrating concepts in text and treats them like other phrases or words.
- Using entire concepts in indexing is highly recommended with SVM, NB and Rocchio with KullbackLeibler. These techniques use concepts in the representation model in order to distinguish classes and improve their effectiveness. The main difference between these techniques and the other variants of Rocchio is that they can employ the distinctive features to distinguish classes, whereas similarity measures used with Rocchio depend on common features to predict classes.

UMLS Concepts and relations among them were used in the other three platforms to evaluate their influence on text classification.

2.2 Semantic enrichment before training

Concerning *Semantic Kernels* (scenario2), results showed deterioration in the performance of Rocchio and its variants after applying Semantic Kernels on vectors that represent corpus documents before training. Our conclusions are as follows:

- *Semantic Kernels* may introduce noise to the text representation and weaken the classifier capability to distinguish classes.
- Introducing similar concepts in text representation in domain specific applications is critical as related concepts may modify the meaning conveyed in the original text.

2.3 Semantic enrichment before prediction

Concerning *Enriching Vectors* (scenario3), the enrichment of text representation in this approach was limited and more disciplined than that *Semantic Kernels*. Here are our conclusions:

- The influence of this enrichment depends on the characteristics of the semantic similarity measure; its basic principle and the range of the values it returns as it is used to modify importance weights of the added features. We recommend using measures that vary between 0 and 1.
- The influence of this enrichment depends also on the similarity measure used in prediction; we recommend using semantic similarities that use common features as the number of these features increases after enrichment.

2.4 Deploying semantics in prediction

Concerning *Semantic Text-To-Text Similarity Measures* (scenario4), they were used in prediction instead of the classical similarity measures of the VSM, test showed better results than the preceding enrichment. Thus, it seems beneficial to Rocchio-based classification to apply Semantic Text-To-Text Similarity Measures for prediction as it modifies the behavior of the classifier and can improve its effectiveness. Note that:

- Weighting scheme used in text representation is essential to these measures and must be taken into account in the aggregation function which is not the case of most of the state of art measures.
- These measures demonstrated less sensitivity on the semantic similarity measures that *Enriching Vectors*. Different semantic similarity measures that varied in principles and ranges of values gave similar results.
- Using Semantic similarity for prediction did not improve the effectiveness of classification as compared with Rocchio using classical similarity measures on the original text or even after conceptualization.

To summarize, involving semantics in text classification process does modify its behavior. Our final conclusions are as follows:

- Conceptualization is the most promising approach as compared to the other three approaches.
- Large classes and low populated classes are those that get maximum attention of the classification technique after involving semantics.
- Semantic Test-to-Test Similarity measures are more adequate to compare vectors of concepts rather than classical similarity measures of the VSM as they take into account the relations between concepts.

3 Perspectives

This thesis focuses on investigating the impact of concepts and their relations on text classification effectiveness through text conceptualization, enriching text representation and using Semantic Test-to-Test Similarity measures in prediction. Our future work is composed of three parts: *short-term*, *medium-term* and *long-term perspectives*.

Concerning *short-term perspectives*, we intent to explore the following points:

- Resolve problems related to scaling and especially for the components that use semantic resources like MetaMap and UMLS::Similarity. New tools for semantic similarity appeared recently and seem very promising. This step is essential to improve the overall *efficiency* in order to enable further investigation.
- Test other families of semantic similarity measures like *IC-based* or *feature based measures*. Experimental study in this restrained evaluation to structure-based measures. Only the range of values returned by these measures was considered as the principle factor in the effectiveness of our implemented approaches. Principles of other families of similarity measures might enhance the effectiveness of our platforms.
- Use other *weighting schemes* for text representation and evaluate their influence on the platform performance compared with TFIDF.
- Evaluate *other scenarios* combining different approaches together. For example, test Test-to-Text semantic similarity measures after enriching text representation using either *Semantic Kernel* or *Enriching Vectors*.
- Test *Conceptualization* on other classification techniques and evaluate its influence on the classification effectiveness using *different strategies*.

Concerning *medium-term perspectives*, we intent to explore the following points:

- Test our approaches on the entire collection of *Ohsumed* or other groups of classes and compare our results with other state of the art approaches. This comparison is necessary, yet complicated as technical details of state of the art works are not completely published or available.
- Test these approaches on other classification, clustering or IR. Many techniques like *KNN* and *K-means* are as extendible as Rocchio and can integrate our approaches.
- Test our approaches on *other collections* related to the medical domain and on real medical text for validation.
- Propose *extensions* to our approaches and test new enriching strategies for text representation enrichment and aggregation function for prediction.
- Combine classifiers built in this work in an *ensemble classifier* in order to improve their effectiveness. For example, we can use a classical classifier on text and a semantic classifier with aggregation function and combine their rankings of predicted classes in order to choose the most appropriate class to a

treated document. This might provide promising ameliorations since it combines the advantages of different classifiers and minimizes their deficiencies.

Concerning *long-term perspective*, we intend to:

- Develop a platform for *indexing medical documents* that enables its users to navigate using thematic classification of these documents. This may be very important and useful for daily work and also for clinical research and health care activity in medical facilities.
- Test our approaches using *general purpose semantic resources* in the medical domain and in other general purpose collections, all towards a *generic platform* for semantic text classification.
- Test our approaches on other types of data like *Web pages* using their metadata, *tweets* and *blogs* from the social networks aiming to establish thematic linking between different sources of information on the Web using *Semantics*.

4 List of Publications

During this thesis, challenges of text classification were published in two research papers in KES2012 and STAIRS2012. Only contributions related to the first points were published in research papers in WI2012 and WISE2012. Moreover, we participated in the Medical Track of TREC2012 using conceptualization and semantic text-to-text similarity measures for ranking. My published research papers are as follows:

- Albitar, S., Espinasse, B., & Fournier, S. (2012). Towards a Supervised Rocchio-based Semantic Classification of Web Pages. *Paper presented at the KES*.
- Albitar, S., Fournier, S., & Espinasse, B. (2012). Towards a Semantic Classifier Committee based on Rocchio. *Paper presented at the STAIRS*.
- Albitar, S., Fournier, S., & Espinasse, B. (2012). Conceptualization Effects on MEDLINE Documents Classification Using Rocchio Method. *Web Intelligence* (pp. 462-466).
- Albitar, S., Fournier, S., & Espinasse, B. (2012). The impact of conceptualization on text classification. *Proceedings of the 13th international conference on Web Information Systems Engineering*, Paphos, Cyprus.
- Hussam Hamdan, Shereen Albitar, Patrice Bellot, Bernard Espinasse, Sébastien Fournier. LSIS at TREC 2012 Medical Track – Experiments with conceptualization, a DFR model and a semantic measure, in : NIST, *The Twenty-First Text REtrieval Conference* (TREC 2012) Notebook, Vol. Special Publication, pp. 12 p., Gaithersburg (USA), nov 2012.
- Bernard Espinasse, Rinaldo Lima, Shereen Albitar, Sébastien Fournier, 'Freitas Fred. Extraction adaptative d'information de pages web par règles d'extraction induites par apprentissage, in : *Revue d'intelligence artificielle*, Vol. 26 (n° 6/2012), pp. 643-678, dec 2012
- Shereen Albitar. Vers une classification sémantique par apprentissage de pages Web basée sur la méthode de Rocchio, *Actes des 8èmes Journées des doctorants du Laboratoires des Sciences de l'Information et des Systèmes J2L6* à Giens, juin 2011.
- Espinasse, B., Fournier, S., Freitas, F., Albitar, S., & Lima, R. (2011). AGATHE-2: An Adaptive, Ontology-Based Information Gathering Multi-Agent System for Restricted Web Domains. In I. Lee (Ed.), *E-Business Applications for Product Development and Competitive Growth: Emerging Technologies* (pp. 236-260). Hershey, PA: Business Science Reference: IGI Global.
- Albitar, S., Espinasse, B., & Fournier, S. (2010). Combining Agents and Wrapper Induction for Information Gathering on Restricted Web Domains. *Paper presented at the Proceedings of the 4th international conference on research challenges in information systems*, RCIS, Nice, France.

REFERENCES

- Aggarwal, C., & Zhai, C. (2012). A Survey of Text Classification Algorithms. In C. C. Aggarwal & C. Zhai (Eds.), *Mining Text Data* (pp. 163-222): Springer US.
- AGROVOC, last access 2013, from <http://aims.fao.org/standards/agrovoc/about>
- Al-Mubaid, H., & Nguyen, H. A. (2006, Aug. 30 2006-Sept. 3 2006). *A Cluster-Based Approach for Semantic Similarity in the Biomedical Domain*. Paper presented at the Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE.
- Albitar, S., Espinasse, B., & Fournier, S. (2010). *Combining Agents and Wrapper Induction for Information Gathering on Restricted Web Domains*. Paper presented at the Proceedings of the 4th international conference on research challenges in information systems, RCIS, Nice, France.
- Albitar, S., Espinasse, B., & Fournier, S. (2012). *Towards a Supervised Rocchio-based Semantic Classification of Web Pages*. Paper presented at the KES. <http://dblp.uni-trier.de/db/conf/kes/kes2012.html#AlbitarEF12>
- Albitar, S., Fournier, S., & Espinasse, B. (2012a). Conceptualization Effects on MEDLINE Documents Classification Using Rocchio Method *Web Intelligence* (pp. 462-466).
- Albitar, S., Fournier, S., & Espinasse, B. (2012b). *The impact of conceptualization on text classification*. Paper presented at the Proceedings of the 13th international conference on Web Information Systems Engineering, Paphos, Cyprus.
- Albitar, S., Fournier, S., & Espinasse, B. (2012c). *Towards a Semantic Classifier Committee based on Rocchio*. Paper presented at the STAIRS. <http://dblp.uni-trier.de/db/conf/stairs/stairs2012.html#AlbitarFE12>
- Apache_Lucene™, last access 2013, from <http://lucene.apache.org/>
- Aristotle, T. b. E. M. E. Categories
- Aronson, A. R. (2001). Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. *Proc AMIA Symp*, 17-21.
- Aronson, A. R., & Lang, F. M. (2010). An overview of MetaMap: historical perspective and recent advances. [Historical Article Research Support, N.I.H., Intramural]. *J Am Med Inform Assoc*, 17(3), 229-236. doi: 10.1136/jamia.2009.002733
- Aronson, A. R., Mork, J. G., Gay, C. W., Humphrey, S. M., & Rogers, W. J. (2004). The NLM Indexing Initiative's Medical Text Indexer. *Stud Health Technol Inform*, 107(Pt 1), 268-272.
- ASCH, V. V. (2012). *DOMAIN SIMILARITY MEASURES: On the use of distance metrics in natural language processing*. Ph.D., Antwerpen university.
- Aseervatham, S., & Bennani, Y. (2009). Semi-structured document categorization with a semantic kernel. *Pattern Recogn.*, 42(9), 2067-2076. doi: 10.1016/j.patcog.2008.10.024
- Astrakhantsev, N. A., & Turdakov, D. Y. (2013). Automatic construction and enrichment of informal ontologies: A survey. *Programming and Computer Software*, 39(1), 34-42. doi: 10.1134/s0361768813010039
- Azuaje, F., Wang, H., & Bodenreider, O. (2005). *Ontology-driven similarity approaches to supporting gene functional assessment*. Paper presented at the Proceedings of the ISMB'2005 SIG meeting on Bio-ontologies.
- Baharudin, B., Lee, L. H., & Khan, K. (2010). A Review of Machine Learning Algorithms for Text-Documents Classification. *Journal of Advances in Information Technology (JAIT)*, 1 (1), 4-20. doi: doi:10.4304/jait.1.1.4-20
- Bai, R., Wang, X., & Liao, J. (2010). *Using an integrated ontology database to categorize web pages*. Paper presented at the Proceedings of the 2010 international conference on Advances in computer science and information technology, Miyazaki, Japan.
- Ball, G. H., & Hall, D. J. (1965). ISODATA. A novel method of data analysis and pattern classification.
- Banerjee, S., & Pedersen, T. (2003). *Extended gloss overlaps as a measure of semantic relatedness*. Paper presented at the Proceedings of the 18th international joint conference on Artificial intelligence, Acapulco, Mexico.

- Bashyam, V., Divita, G., Bennett, D. B., Browne, A. C., & Taira, R. K. (2007). A normalized lexical lookup approach to identifying UMLS concepts in free text. [Research Support, N.I.H., Extramural]. *Stud Health Technol Inform*, 129(Pt 1), 545-549.
- Berry, M. W., Dumais, S. T., & O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. *SIAM Rev.*, 37(4), 573-595. doi: 10.1137/1037127
- Bhatia, N., Shah, N. H., Rubin, D. L., Chiang, A. P., & Musen, M. A. (2008). *Comparing Concept Recognizers for Ontology-Based Indexing: MGREP vs. MetaMap*.
- BioPortal, last access 2013, from <http://bioportal.bioontology.org/>
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 993-1022.
- Bloehdorn, S., & Hotho, A. (2006). *Boosting for text classification with semantic features*. Paper presented at the Proceedings of the 6th international conference on Knowledge Discovery on the Web: advances in Web Mining and Web Usage Analysis, Seattle, WA.
- Bloehdorn, S., & Moschitti, A. (2007). *Combined syntactic and semantic Kernels for text classification*. Paper presented at the Proceedings of the 29th European conference on IR research, Rome, Italy.
- Borko, H., & Bernick, M. (1963). Automatic Document Classification. *J. ACM*, 10(2), 151-162. doi: 10.1145/321160.321165
- Boubekeur, F. (2008). *Contribution à la définition de modèles flexibles de recherche d'information basés sur les CP-Nets*. Ph.D., Université Paul Sabatier.
- Bulskov, H., Knappe, R., & Andreasen, T. (2002). *On Measuring Similarity for Conceptual Querying*. Paper presented at the Proceedings of the 5th International Conference on Flexible Query Answering Systems.
- Burges, C. J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Min. Knowl. Discov.*, 2(2), 121-167. doi: 10.1023/a:1009715923555
- Cambridge Dictionaries Online, Cambridge University Press last access 2013, from <http://dictionary.cambridge.org/dictionary/american-english/>
- Caropreso, M. F., Matwin, S., & Sebastiani, F. (2001). A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In A. G. Chin (Ed.), *Text databases & document management* (pp. 78-102): IGI Publishing
- Caviedes, J. E., & Cimino, J. J. (2004). Towards the development of a conceptual distance metric for the UMLS. *J. of Biomedical Informatics*, 37(2), 77-85. doi: 10.1016/j.jbi.2004.02.001
- Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3), 1-27. doi: 10.1145/1961189.1961199
- Chirita, P. A., Nejdl, W., Paiu, R., Kohlsch, C., #252, & tter. (2005). *Using ODP metadata to personalize search*. Paper presented at the Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, Salvador, Brazil.
- Cohen, P. R., & Kjeldsen, R. (1987). Information retrieval by constrained spreading activation in semantic networks. *Information Processing & Management*, 23(4), 255-268. doi: [http://dx.doi.org/10.1016/0306-4573\(87\)90017-3](http://dx.doi.org/10.1016/0306-4573(87)90017-3)
- Crossno, P. J., Wilson, A. T., Shead, T. M., & Dunlavy, D. M. (2011). *TopicView: Visually Comparing Topic Models of Text Collections*. Paper presented at the Proceedings of the 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence.
- Cycorp. Home of smarter solutions, last access 2013, from <http://www.cyc.com/platform/opencyc>
- Dai, M. (2008). *An Efficient Solution for Mapping Free Text to ontology Terms*. Paper presented at the AMIA Summit on Translational Bioinformatics, San Francisco, CA.
- Daoud, M. (2009). *Accès personnalisé à l'information : approche basée sur l'utilisation d'un profil utilisateur sémantique dérivé d'une ontologie de domaines à travers l'historique des sessions de recherche*. Ph.D., Université Paul Sabatier - Toulouse III.
- Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by Latent Semantic Analysis. *JASIS*, 41(6), 391-407.

- Deveaud, R., Bonnefoy, L., & Bellot, P. (2013, 3-5 april). *Quantification et identification des concepts implicites d'une requête*. Paper presented at the CORIA, Neuchâtel.
- Dewey, M. (2011). *Dewey Decimal Classification and Relative Index*: Oclc; 23 edition (May 2011).
- Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7), 1895-1923. doi: 10.1162/089976698300017197
- Dinh, D., & Tamine, L. (2012). Towards a context sensitive approach to searching information based on domain specific knowledge sources. *Web Semantics: Science, Services and Agents on the World Wide Web*, 12-13(0), 41-52. doi: <http://dx.doi.org/10.1016/j.websem.2011.11.009>
- Dobrev, M., Gocheva, D., & Batchkova, I. (2008, 6-8 Sept. 2008). *An ontological approach for planning and scheduling in primary steel production*. Paper presented at the Intelligent Systems, 2008. IS '08. 4th International IEEE Conference.
- Dolan, B., Quirk, C., & Brockett, C. (2004). *Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources*. Paper presented at the Proceedings of the 20th international conference on Computational Linguistics, Geneva, Switzerland.
- Duan, K.-B., & Keerthi, S. S. (2005). Which Is the Best Multiclass SVM Method? An Empirical Study. In N. Oza, R. Polikar, J. Kittler & F. Roli (Eds.), *Multiple Classifier Systems* (Vol. 3541, pp. 278-285): Springer Berlin Heidelberg.
- Dubin, D. (2004). The most influential paper Gerard Salton never wrote. *Library trends*, 52(4), 748-764.
- EL-Manzalawy, Y., & Honavar, V. (2005). WLSVM: Integrating LibSVM into Weka Environment. Software available at <http://www.cs.iastate.edu/~yasser/wlsvm>.
- Espinasse, B., Fournier, S., Freitas, F., Albitar, S., & Lima, R. (2011). AGATHE-2: An Adaptive, Ontology-Based Information Gathering Multi-Agent System for Restricted Web Domains. In I. Lee (Ed.), *E-Business Applications for Product Development and Competitive Growth: Emerging Technologies* (pp. 236-260). Hershey, PA: Business Science Reference: IGI Global.
- Everitt, B. S. (1992). *The Analysis of Contingency Tables, Second Edition* (2nd edition ed.): Chapman and Hall/CRC
- Ferretti, E., Errecalde, M., & Rosso, P. (2008). Does Semantic Information Help in the Text Categorization Task? *Journal of Intelligent Systems*, 17, 91-107.
- Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., & Ruppín, E. (2002). Placing search in context: the concept revisited. *ACM Transaction on Information Systems*, 20(1), 116-131. doi: 10.1145/503104.503110
- Gabrilovich, E., & Markovitch, S. (2007). *Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis*. Paper presented at the Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India.
- Gabrilovich, E., & Markovitch, S. (2009). Wikipedia-based semantic interpretation for natural language processing. *J. Artif. Int. Res.*, 34(1), 443-498.
- Geng, X., Liu, T.-Y., Qin, T., & Li, H. (2007). *Feature selection for ranking*. Paper presented at the Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, Amsterdam, The Netherlands.
- Girju, R., Nakov, P., Nastase, V., Szpakowicz, S., Turney, P., & Yuret, D. (2007). *SemEval-2007 task 04: classification of semantic relations between nominals*. Paper presented at the Proceedings of the 4th International Workshop on Semantic Evaluations, Prague, Czech Republic.
- Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, 43(5-6), 907-928. doi: 10.1006/ijhc.1995.1081
- Guisse, A., Khelif, K., & Collard, M. (2009). *PatClust : une plateforme pour la classification sémantique des brevets*. Paper presented at the Conférence d'Ingénierie des connaissances, Hammamet, Tunisie.
- Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3(3/1/2003), 1157--1182.

- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1), 10-18. doi: 10.1145/1656274.1656278
- Han, E.-H., & Karypis, G. (2000). *Centroid-Based Document Classification: Analysis and Experimental Results*. Paper presented at the 4th European Conference on Principles of Data Mining and Knowledge Discovery.
- Hao, T., Lu, Z., Wang, S., Zou, T., GU, S., & Wenxin, L. (2008). *Categorizing and ranking search engine's results by semantic similarity*. Paper presented at the Proceedings of the 2nd international conference on Ubiquitous information management and communication, Suwon, Korea.
- Hartmann, R. R. K., & James, G. (1998). *Dictionary of Lexicography*. London: Routledge.
- Hersh, W., Buckley, C., Leone, T. J., & Hickam, D. (1994). *OHSUMED: an interactive retrieval evaluation and new large test collection for research*. Paper presented at the 17th annual international ACM SIGIR conference on Research and development in information retrieval, Dublin, Ireland.
- Hliaoutakis, A. (2005). *semantic similarity measures in mesh ontology and their application to information retrieval on medline*. Technical University of Crete.
- Hliaoutakis, A., Varelas, G., Petrakis, E. M., & Milios, E. (2006). MedSearch: A Retrieval System for Medical Information Based on Semantic Similarity. In J. Gonzalo, C. Thanos, M. F. Verdejo & R. Carrasco (Eds.), *Research and Advanced Technology for Digital Libraries* (Vol. 4172, pp. 512-515): Springer Berlin Heidelberg.
- Hofmann, T. (1999). *Probabilistic latent semantic indexing*. Paper presented at the Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, Berkeley, California, USA.
- Hotho, A., Staab, S., & Stumme, G. (2003). Text clustering based on background knowledge.
- Huang, A. (2008). *Similarity measures for text document clustering*. Paper presented at the Sixth New Zealand Computer Science Research Student Conference, , Christchurch, New Zealand.
- Huang, L. (2011). *Concept-based text clustering*. Doctor of Philosophy in Computer Science, University of Waikato.
- Huang, L., Milne, D., Frank, E., & Witten, I. H. (2012). Learning a concept-based document similarity measure. *J. Am. Soc. Inf. Sci. Technol.*, 63(8), 1593-1608. doi: 10.1002/asi.22689
- Jiang, J. J., & Conrath, D. W. (1997). *Semantic similarity based on corpus statistics and lexical taxonomy*. Paper presented at the Proc. of the Int'l. Conf. on Research in Computational Linguistics.
- Joachims, T. (1998). *Text categorization with support vector machines: learning with many relevant features*. Paper presented at the Proceedings of ECML-98, 10th European Conference on Machine Learning, Chemnitz, Germany.
- Knappe, R., Bulskov, H., & Andreassen, T. (2007). Perspectives on ontology-based querying: Research Articles. *Int. J. Intell. Syst.*, 22(7), 739-761. doi: 10.1002/int.v22:7
- Kuncheva, L. I. (2004). *Combining Pattern Classifiers: Methods and Algorithms*.
- Lan, M., Tan, C. L., Su, J., & Lu, Y. (2009). Supervised and Traditional Term Weighting Methods for Automatic Text Categorization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(4), 721-735. doi: 10.1109/tpami.2008.110
- Leacock, C., & Chodorow, M. (1998). Combining Local Context and WordNet Similarity for Word Sense Identification. In C. Fellbaum (Ed.), *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)* (pp. 265-283): The MIT Press.
- Lee, M., Pincombe, B., & Welsh, M. (2005). An Empirical Evaluation of Models of Text Document Similarity *Proceedings of the 27th Annual Conference of the Cognitive Science Society* (pp. 1254-1259): Erlbaum.
- Lesk, M. (1986). *Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone*. Paper presented at the Proceedings of the 5th annual international conference on Systems documentation, Toronto, Ontario, Canada.

- Lewis, D. D. (1998). *Naïve (Bayes) at Forty: The Independence Assumption in Information Retrieval*. Paper presented at the Proceedings of the 10th European Conference on Machine Learning.
- Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). RCV1: A New Benchmark Collection for Text Categorization Research. *J. Mach. Learn. Res.*, 5, 361-397.
- Li, Y., Bandar, Z. A., & McLean, D. (2003). An approach for measuring semantic similarity between words using multiple information sources. *Knowledge and Data Engineering, IEEE Transactions on*, 15(4), 871-882. doi: 10.1109/tkde.2003.1209005
- Li, Y., Chung, S. M., & Holt, J. D. (2008). Text document clustering based on frequent word meaning sequences. *Data Knowl. Eng.*, 64(1), 381-404. doi: 10.1016/j.datak.2007.08.001
- Li, Z., Li, P., Wei, W., Liu, H., He, J., Liu, T., & Du, X. (2009). AutoPCS: A Phrase-Based Text Categorization System for Similar Texts. In Q. Li, L. Feng, J. Pei, S. Wang, X. Zhou & Q.-M. Zhu (Eds.), *Advances in Data and Web Management* (Vol. 5446, pp. 369-380): Springer Berlin / Heidelberg.
- Lin, D. (1998). *An Information-Theoretic Definition of Similarity*. Paper presented at the Proceedings of the Fifteenth International Conference on Machine Learning.
- Liu, T., Chen, Z., Zhang, B., Ma, W.-y., & Wu, G. (2004). *Improving Text Classification using Local Latent Semantic Indexing*. Paper presented at the Proceedings of the Fourth IEEE International Conference on Data Mining.
- MacQueen, J. (1967). *Some methods for classification and analysis of multivariate observations*. Paper presented at the Proc. Fifth Berkeley Symp. on Math. Statist. and Prob.
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.
- Mao, W., & Chu, W. W. (2002). Free-text medical document retrieval via phrase-based vector space model. *Proc AMIA Symp*, 489-493.
- McInnes, B. T., Pedersen, T., Liu, Y., Melton, G. B., & Pakhomov, S. V. (2011). *Knowledge-based Method for Determining the Meaning of Ambiguous Biomedical Terms Using Information Content Measures of Similarity*. Paper presented at the In Proceedings of the American Medical Informatics Symposium
- McInnes, B. T., Pedersen, T., & Pakhomov, S. V. S. (2009). *UMLS-Interface and UMLS-Similarity : Open Source Software for Measuring Paths and Semantic Similarity*. Paper presented at the Proceedings of the Annual Symposium of the American Medical Informatics Association, San Francisco, CA.
- Medical Subject Headings, last access 2013, from <http://www.nlm.nih.gov/pubs/factsheets/mesh.html>
- Meystre, S. M., Thibault, J., Shen, S., Hurdle, J. F., & South, B. R. (2010). Texttractor: a hybrid system for medications and reason for their prescription extraction from clinical text documents. [Research Support, N.I.H., Extramural]. *J Am Med Inform Assoc*, 17(5), 559-562. doi: 10.1136/jamia.2010.004028
- Mihalcea, R. (2007). *Using Wikipedia for Automatic Word Sense Disambiguation*. Paper presented at the North American Chapter of the Association for Computational Linguistics (NAACL 2007).
- Mihalcea, R., Corley, C., & Strapparava, C. (2006). *Corpus-based and knowledge-based measures of text semantic similarity*. Paper presented at the Proceedings of the 21st national conference on Artificial intelligence - Volume 1, Boston, Massachusetts.
- Miller, G. A. (1995). WordNet: a lexical database for English. *Commun. ACM*, 38(11), 39-41. doi: 10.1145/219717.219748
- Milne, D., & Witten, I. H. (2008). *Learning to Link with Wikipedia*. Paper presented at the Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, California, USA.
- Milne, D. N., Witten, I. H., & Nichols, D. M. (2007). *A Knowledge-based Search Engine Powered by Wikipedia*. Paper presented at the Proceedings of the Sixteenth ACM

- Conference on Conference on Information and Knowledge Management, Lisbon, Portugal.
- Mitra, V., Wang, C.-J., & Banerjee, S. (2007). Text classification: A least square support vector machine approach. *Applied Soft Computing*, 7(3), 908-914. doi: 10.1016/j.asoc.2006.04.002
- Mohler, M., & RadaMihalcea. (2009). *Text-to-text semantic similarity for automatic short answer grading*. Paper presented at the Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, Athens, Greece.
- Movie Review Data, last access 2013, from <http://www.cs.cornell.edu/people/pabo/movie-review-data/>
- Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2), 1-69. doi: 10.1145/1459352.1459355
- Open Directory Project, last access 2013, from <http://www.dmoz.org/>
- Oxford Dictionary of Statistic, last access 2013, from <http://www.answers.com/library/Statistics+Dictionary-cid-25353924>
- Özgür, A., Özgür, L., & Güngör, T. (2005). *Text categorization with class-based and corpus-based keyword selection*. Paper presented at the Proceedings of the 20th international conference on Computer and Information Sciences, Istanbul, Turkey.
- Patwardhan, S., & Pedersen, T. (2006). *Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts*. Paper presented at the EACL 2006 Workshop Making Sense of Sense---Bringing Computational Linguistics and Psycholinguistics Together.
- Pedersen, T., Pakhomov, S., McInnes, B., & Liu, Y. (2012). *Measuring the Similarity and Relatedness of Concepts in the Medical Domain*. Paper presented at the 2nd ACM SIGHIT International Health Informatics Symposium (IHI 2012), Miami, Florida.
- Pedersen, T., Patwardhan, S., & Michelizzi, J. (2004). *WordNet::Similarity: measuring the relatedness of concepts*. Paper presented at the Demonstration Papers at HLT-NAACL 2004, Boston, Massachusetts.
- Peng, X., & Choi, B. (2005). *Document classifications based on word semantic hierarchies*. Paper presented at the International Conference on Artificial Intelligence and Applications (AIA'05).
- Petrakis, E. G. M., Varelas, G., Hliaoutakis, A., & Raftopoulou, P. (2006). X-Similarity: Computing Semantic Similarity between Concepts from Different Ontologies. *Journal of Digital Information Management (JDIM)*, 4.
- Pierre, J. M. (2001). On the Automated Classification of Web Sites. *Linköping Electronic Articles in Computer and Information Science*, 6(1).
- Pirro, G. (2009). A semantic similarity metric combining features and intrinsic information content. *Data Knowl. Eng.*, 68(11), 1289-1308. doi: 10.1016/j.datak.2009.06.008
- Pirro, G., & Euzenat, J. (2010). *A feature and information theoretic framework for semantic similarity and relatedness*. Paper presented at the Proceedings of the 9th international semantic web conference on The semantic web - Volume Part I, Shanghai, China.
- Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130--137.
- Prabowo, R., Jackson, M., Burden, P., & Knoell, H.-D. (2002). *Ontology-Based Automatic Classification for the Web Pages: Design, Implementation and Evaluation*. Paper presented at the Proceedings of the 3rd International Conference on Web Information Systems Engineering.
- PubMed Tutorial, last access 2013, from <http://www.nlm.nih.gov/bsd/disted/pubmedtutorial/>
- Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and application of a metric on semantic nets. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(1), 17-30. doi: 10.1109/21.24528
- Renard, A., Calabretto, S., & Rumpler, B. (2011). Towards a Better Semantic Matching for Indexation Improvement of Error-Prone (Semi-)Structured XML Documents. In J. Filipe & J. Cordeiro (Eds.), *Web Information Systems and Technologies* (Vol. 75, pp. 286-298): Springer Berlin Heidelberg.

- Home Page for 20 Newsgroups Data Set, last access 2013, from <http://people.csail.mit.edu/jrennie/20Newsgroups>
- Resnik, P. (1995). *Using information content to evaluate semantic similarity in a taxonomy*. Paper presented at the Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1, Montreal, Quebec, Canada.
- Ruch, P., Gobeill, J., Lovis, C., & Geissbühler, A. (2008). Automatic medical encoding with SNOMED categories. *BMC Medical Informatics and Decision Making*, 8(1), 1-8. doi: 10.1186/1472-6947-8-s1-s6
- Rus, V., Lintean, M., Banjade, R., Niraula, N., & Stefanescu, D. (2013). *SEMILAR: The Semantic Similarity Toolkit*. Paper presented at the Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria.
- Salton, G. (1971). *The SMART Retrieval System-Experiments in Automatic Document Processing*: Prentice-Hall, Inc.
- Salton, G. (1989). *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Salton, G., & Buckley, C. (1988). *On the use of spreading activation methods in automatic information*. Paper presented at the Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval, Grenoble, France.
- Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM*, 18(11), 613-620. doi: 10.1145/361219.361220
- Sanchez, D., & Batet, M. (2011). Semantic similarity estimation in the biomedical domain: An ontology-based information-theoretic perspective. *J. of Biomedical Informatics*, 44(5), 749-759. doi: 10.1016/j.jbi.2011.03.013
- Sanchez, D., Batet, M., & Isern, D. (2011). Ontology-based information content computation. *Know.-Based Syst.*, 24(2), 297-303. doi: 10.1016/j.knosys.2010.10.001
- Sanchez, D., Batet, M., Isern, D., & Valls, A. (2012). Ontology-based semantic similarity: A new feature-based approach. *Expert Syst. Appl.*, 39(9), 7718-7728. doi: 10.1016/j.eswa.2012.01.082
- Séaghdha, D. Ó. (2009). *Semantic classification with WordNet kernels*. Paper presented at the Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, Boulder, Colorado.
- Séaghdha, D. Ó., & Copestake, A. (2008). *Semantic Classification with Distributional Kernels*. Paper presented at the Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008).
- Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1), 1-47. doi: 10.1145/505282.505283
- Sebastiani, F. (2005). Text Categorization *Encyclopedia of Database Technologies and Applications* (pp. 683-687): Idea Group.
- Seco, N., Veale, T., & Hayes, J. (2004, 2004). *An Intrinsic Information Content Metric for Semantic Similarity in WordNet*. Paper presented at the ECAI.
- Semantic Measures Library & ToolKit, last access 2013, from <http://www.semantic-measures-library.org/sml/index.php>
- Shah, N. H., Bhatia, N., Jonquet, C., Rubin, D., Chiang, A. P., & Musen, M. A. (2009). Comparison of concept recognizers for building the Open Biomedical Annotator. [Comparative Study Research Support, N.I.H., Extramural]. *BMC Bioinformatics*, 10 Suppl 9, S14. doi: 10.1186/1471-2105-10-S9-S14
- Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427-437. doi: 10.1016/j.ipm.2009.03.002

- Somasundaram, K., & Murphy, G. C. (2012). *Automatic categorization of bug reports using latent Dirichlet allocation*. Paper presented at the Proceedings of the 5th India Software Engineering Conference, Kanpur, India.
- Soucy, P., & Mineau, G. W. (2001). *A Simple KNN Algorithm for Text Categorization*. Paper presented at the Proceedings of the 2001 IEEE International Conference on Data Mining.
- Stavrianou, A., Andritsos, P., & Nicoloyannis, N. (2007). Overview and semantic issues of text mining. *SIGMOD Rec.*, 36(3), 23-34. doi: 10.1145/1324185.1324190
- Stein, B., Eissen, S. M. z., & Potthast, M. (2006). *Syntax versus semantics: Analysis of enriched vector space models*. Paper presented at the Third International Workshop on Text-Based Information Retrieval (TIR 06), University of Trento, Italy.
- Suggested Upper Merged Ontology, last access 2013, from <http://www.ontologyportal.org/>
- Taghva, K., Borsack, J., Coombs, J., Condit, A., Lumos, S., & Nartker, T. (2003). *Ontology-based Classification of Email*. Paper presented at the Proceedings of the International Conference on Information Technology: Computers and Communications.
- Text REtrieval Conference, last access 2013, from <http://trec.nist.gov/>
- Trillo, R., Po, L., Ilarri, S., Bergamaschi, S., & Mena, E. (2011). Using semantic techniques to access web data. *Inf. Syst.*, 36(2), 117-133. doi: 10.1016/j.is.2010.06.008
- Tversky, A. (1977). Features of similarity. *Psychological Review*, 84, 327--352.
- Unified Medical Language System, last access 2013, from <http://www.nlm.nih.gov/research/umls/>
- University of Minnesota Pharmacy Informatics Lab, last access 2013, from <http://rxinformatics.umn.edu/SemanticRelatednessResources.html>
- Vapnik, V. (1998). *Statistical learning theory*. NY: Springer-Verlag.
- Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc.
- Wang, J. Z., & Taylor, W. (2007). *Concept Forest: A New Ontology-assisted Text Document Similarity Measurement Method*. Paper presented at the Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence.
- Wang, P., & Domeniconi, C. (2008). *Building semantic kernels for text classification using wikipedia*. Paper presented at the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, Las Vegas, Nevada, USA.
- Wang, P., Hu, J., Zeng, H.-J., Chen, L., & Chen, Z. (2007). *Improving Text Classification by Using Encyclopedia Knowledge*. Paper presented at the Proceedings of the 2007 Seventh IEEE International Conference on Data Mining.
- Wikipedia:About. From Wikipedia, the free encyclopedia., last access 2013, from <http://en.wikipedia.org/wiki/Wikipedia:About>
- Wongthongtham, P., Chang, E., Dillon, T., & Sommerville, I. (2009). Development of a Software Engineering Ontology for Multisite Software Development. *Knowledge and Data Engineering, IEEE Transactions on*, 21(8), 1205-1217. doi: 10.1109/tkde.2008.209
- Wu, Z., & Palmer, M. (1994). *Verbs semantics and lexical selection*. Paper presented at the Proceedings of the 32nd annual meeting on Association for Computational Linguistics, Las Cruces, New Mexico.
- YAGO2s: A High-Quality Knowledge Base, last access 2013, from <http://www.mpi-inf.mpg.de/yago-naga/yago/>
- Yang, Y., & Liu, X. (1999). *A re-examination of text categorization methods*. Paper presented at the Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, Berkeley, California, USA.
- Yang, Y., & Pedersen, J. O. (1997). *A Comparative Study on Feature Selection in Text Categorization*. Paper presented at the Proceedings of the Fourteenth International Conference on Machine Learning.
- Yi, K., & Beheshti, J. (2009). A hidden Markov model-based text classification of medical documents. *J. Inf. Sci.*, 35(1), 67-81. doi: 10.1177/0165551508092257

- Zhong, J., Zhu, H., Li, J., & Yu, Y. (2002). *Conceptual Graph Matching for Semantic Search*. Paper presented at the Proceedings of the 10th International Conference on Conceptual Structures: Integration and Interfaces.
- Zhou, X., Zhang, X., & Hu, X. (2006). *MaxMatcher: biological concept extraction using approximate dictionary lookup*. Paper presented at the Proceedings of the 9th Pacific Rim international conference on Artificial intelligence, Guilin, China.
- Zhou, Z., Wang, Y., & Gu, J. (2008). *A New Model of Information Content for Semantic Similarity in WordNet*. Paper presented at the Proceedings of the 2008 Second International Conference on Future Generation Communication and Networking Symposia - Volume 03.
- Zhu, S., Zeng, J., & Mamitsuka, H. (2009). Enhancing MEDLINE document clustering by incorporating MeSH semantic similarity. *Bioinformatics*, 25(15), 1944-1951. doi: 10.1093/bioinformatics/btp338
- Ziegler, P., Kiefer, C., Sturm, C., Dittrich, K. R., & Bernstein, A. (2006). *Generic similarity detection in ontologies with the SOQA-SimPack toolkit*. Paper presented at the Proceedings of the 2006 ACM SIGMOD international conference on Management of data, Chicago, IL, USA.