École doctorale n° 224 : Cognition, Langage, Interaction

# THÈSE

**pour obtenir le grade de docteur délivré par**

## Université Paris 8

### Spécialité doctorale “Informatique”

*présentée et soutenue publiquement par*

**Hosni SEFFIH**

le 17 décembre 2020

# Détection d’événements géo-chrono-localisés dans les tweets

Directrice de thèse : **Myriam LAMOLLE**
Co-encadrant de thèse : **Christian FLUHR**

**Jury**

| | | |
|---|---|---|
| **Sylvie DESPRES,** | Professeure, Université Paris 13 | Rapportrice |
| **Patrick PAROUBEK,** | Ingénieur Recherche, LIMSI-CNRS | Rapporteur |
| **Abderrahman EL MHAMEDI,** | Professeur, Université Paris 8 | Examinateur |
| **Ernesto EXPOSITO,** | Professeur, Université de Pau et des Pays de l’Adour | Examinateur |

**Université Paris 8**
**Laboratoire d’informatique avancée de Saint-Denis (LIASD)**
**IUT de Montreuil**
**Saint-Denis, France**

THESE

# Remerciements

J'aimerais tout d'abord remercier ma directrice de thèse, Professeure Myriam Lamolle, pour m'avoir accueilli au sein de son équipe grâce à un contrat CIFRE me permettant de mener à bien mon avenir doctoral. Un grand merci pour son accompagnement tout au long de ces trois années ainsi que pour sa relecture finale méticuleuse de chacun des chapitres de cette thèse.

J'adresse aussi mes chaleureux remerciements à mon co-encadrant de thèse, M. Christian Fluhr, pour son attention de tous les instants sur mes travaux, ainsi que pour ses compétences scientifiques, ses conseils avisés, sa franchise et sa sympathie. J'ai beaucoup appris à ses côtés et je lui adresse ma gratitude pour tout cela.

Je remercie aussi Aurélie Pradelles, la responsable d'équipe chez GEOLSemantics, pour m'avoir accompagné tout au long de ce travail, pour ses corrections et ses mises en perspective pertinentes entre la recherche et l'ingénierie. Dorénavant, dans ma tête, le mot *compétent* est synonyme d'Aurélie. Je remercie aussi mes collègues très « auréliens » : Houda, Zhen, Mehdi, Stéphane et Jérémy.

Je tiens également à remercier Mme Sylvie Despres, professeure à l'Université Paris 13, et M. Patrick Paroubek, ingénieur de recherche au LIMSI, qui m'ont fait l'honneur d'accepter d'être rapporteurs de cette thèse.

J'associe à ces remerciements M. Abderrahman EL MHAMEDI, professeur à l'Université Paris 8, et M. Ernesto EXPOSITO, professeur à l'Université de Pau et des Pays de l'Adour, pour avoir accepté d'examiner mon travail.

Je désire en outre remercier tous mes collègues et amis de l'IUT de Montreuil pour leur bonne humeur et leur sympathie.

Ces remerciements seraient incomplets si je n'en adressais pas à ma famille dont l'encouragement et le soutien constants m'ont été d'un très grand réconfort. Essayer de trouver les mots simples afin d'expliquer ma spécialité à ma mère de 74 ans était pour moi un exercice afin d'évaluer le degré de maîtrise de mon sujet : plus je me familiarisais avec la problématique, plus mes exemples étaient simples et compréhensibles par elle. Je tiens à la remercier pour ça et bien plus encore.

Et pour finir je tiens à remercier ma vie personnelle de ne pas avoir existé afin que je puisse finir ce travail dans les temps...

# Résumé

La détection d'événements à partir de textes est un défi encore aujourd'hui. En effet, les sources de données textuelles sont plus diversifiées avec les nouvelles technologies de communication, en particulier dans leur mode de rédaction. Dans le contexte des réseaux sociaux, les informations sont souvent écrites dans un style particulier en modifiant les caractéristiques orthographiques et/ou grammaticales afin de réduire la longueur du texte car le nombre de caractères autorisé est limité. De plus, il est possible d'utiliser des formalismes particuliers comme les mots-dièses (dits *hashtags*) qui mettent en exergue un thème particulier. Tout cet environnement particulier augmente la difficulté de détection d'événements. Lors de ces travaux de recherche, nous avons cherché à extraire des événements géo-chrono-localisés dans des tweets (plus particulièrement sur la ville de Nice - France pour nos expérimentations).

Pour arriver à nos fins, nous devons faire face à trois verrous scientifiques à savoir (i) la constitution d'un corpus pertinent sur un territoire particulier en regard d'une liste d'événements, (ii) le traitement sémantique des réseaux sociaux et plus particulièrement Twitter, (iii) la détection des éléments qui composent un événement afin de lever toute ambiguïté sur son importance car il sera déclencheur d'alerte.

Le système proposé, appellé GEOLsig, passe par la gestion du flux des tweets (récupération de tweets + prétraitement de l'orthographe) puis transformation du texte brut des tweets en texte structuré afin de passer à la phase d'extraction de l'information des lieux, des dates et des actions. Ce sont les composants constitutifs de nos événements. L'information concernant le lieu est vérifiée sur une base de données géographique qui confirme ou non la véracité du lieu et renvoie une géolocalisation qui va servir à situer l'événement sur une carte, utile pour les services publics qui reçoivent l'alerte.

Afin de constituer un corpus sur un territoire particulier, dans le cas de Twitter et dans le respect des règles de la RGPD, nous avons lancé des requêtes contenant des lieux (pour l'expérimentation sur Nice) par l'application Twint. Pour les associer à des événements, nous avons ajouté à la requête des verbes et des termes en relation avec un événement donné ; par exemple, "Nécropole de Nice" AND ((feu OR incendie OR flamme) -coup -arme -artifice). Ceci nous a permis d'avoir un corpus suffisamment varié pour entraîner notre outil d'extraction d'événements lors de la phase d'apprentissage.

Quant à notre deuxième problématique concernant le traitement sémantique de tweets, la solution de GEOLSemantics gèrant des textes propres qui respectent les règles orthographiques du français n'était pas exploitable vu la nature des tweets. De ce fait, nous avons

dû passer par une phase de pré-traitement par lequel ces derniers sont transformés en utilisant un dictionnaire argotique. Par exemple, « tkt » est remplacé par « Ne t'inquiète pas ». Un outil de phonétisation a été développé aussi afin de corriger les fautes d'orthographe les plus communes dû à la prononciation des mots. Une fois les tweets transformés, une analyse linguistique est réalisée afin de transformer le texte brut de ces tweets en texte structuré. Notre système se fonde sur une analyse linguistique profonde nommée GEOL-Analyseur-Linguistique (GAL), réalisée grâce à un moteur développé en interne et qui repose sur des modules de tokenisation, de lemmatisation et de désambiguïsation.

Enfin, pour lever le troisième verrou scientifique à savoir la détection des éléments qui composent un événement, nous avons mis en place un système d'extraction (GEX) d'information à partir des données textuelles produites par GAL. Pour ce faire, GEX s'appuie sur une ontologie, des ressources linguistiques, un exécuteur de règles d'extraction (dit ERG) et un module de mise en cohérence (dit GEXIH).

Les lieux détectés sont ensuite filtrés par une base de lieux géographiques qui permet de renvoyer pour chaque lieu existant sa géolocalisation précise servant à afficher l'événement sur une carte.

Afin de vérifier la pertinence du processus complet des traitements linguistiques effectués, nous avons mené des expérimentations sur différents corpus de tweets. Ces expérimentations ont montré le bien fondé de coupler les traitements linguistiques par règle avec les techniques d'apprentissage (ou *machine learning*), dans notre cas par l'utilisation d'un réseau de neurones *BERT*.

Nos futurs travaux aborderont à court terme la détection automatique implémenté par un bot sur Twitter, et à plus long terme, (i) l'optimisation du couplage BERT / traitements linguistiques et sémantiques GEOL, (ii) le traitement d'autres langues, (iii) l'extension de la solution à d'autres réseaux sociaux.

*Mots-clés : traitement linguistique, extraction sémantique, réseaux sociaux, détection d'événements, système d'alerte*

# Abstract

Events detecting in texts is still challenging today. The new communication technologies brought more diversity to the textual data sources, in particular in their way of writing. In the context of social networks, written information is often conveyed in a simplified manner. This makes event detection more difficult. In this thesis, we sought to extract geo-chrono-localized events in tweets. We focused the experiments area using the city of Nice in France.

To achieve our goals, we must face three main issues, namely (i) the establishment of a corpora on a given region with regard to a list of events, (ii) the semantic processing of social networks and more particularly Twitter, (iii) detecting the elements making up an event in order to remove any ambiguity about its relevance. Indeed, this event will potentially trigger an alert.

The proposed system is called GEOLsig. The GEOLsig process involves managing the flow of tweets (extractng Twitter data + preprocessing). Then, the next step consists in transformating tweets raw text into structured text. This is followed by a step of extracting information about the places, dates and actions. These are the components of our events. The information on the place is checked on a geographic database which confirms or not the veracity of the place and returns a geolocation which will be used to display the event on a map.

In order to constitute a corpora for a particular region in the case of Twitter, and in compliance with the RGPD rules, we launched requests containing places (for experimentation in Nice) using the Twint API. To link them with events, we added verbs and terms related to a given event to the query ; for example, “Nécropole de Nice” AND ((fire OR fire OR flame) -coup -arme -artifice). This allowed us to have a sufficiently varied corpus to train our event extraction tool during the step of learning machine.

Semantic processing of tweets is the second issue. The GEOLSemantics tool being developed on clean texts which respect the spelling rules of French did not perform well enough as is because of the way of tweet writing. Therefore, we had to go through a pre-processing phase by which they are transformed using a slang dictionary. For example, "tkt" is replaced by "Ne t’inquiète pas". A phonetization tool has also been developed to correct the most common spelling errors due to the pronunciation of words. Once the tweets have been transformed, they go through a linguistic analysis in order to transform the plain text into structured text. Our system is based on a deep linguistic analysis called GEOL-Analyzer-Linguistique (GAL), carried out thanks to an engine developed in-house. This engine is based on tokenization, lemmatization and disambiguation modules.

Finally, to remove the third issue, namely the detection of the elements that make up an event, we have set up an information extraction system (GEX) from the textual data produced by GAL. It is based on an ontology, linguistic resources, a generator of extraction rules (called ERG), and a module for consistency (called GEXIH).

The detected places are then filtered using geographical places database, which makes it possible to return for each existing place its precise geolocation used to display the event on a map.

Different experiments on different corpora will help us to test our linguistic processing relevance.

Our future work will address the automatic detection implemented by a bot on Twitter in the short term. And in the longer term, it will address (i) optimization of the BERT / GEOL semantic linguistic processing, (ii) processing other languages, (iii) extending the solution to other social networks.

*Keywords : semantic linguistic processing, social networks, event detection, alert system*

# Liste des publications

La plupart de nos résultats ont été publiés. Nous listons ici celles associées à cette thèse.

1. Hosni Seffih, Myriam Lamolle, Aurélie Pradelles, Zhen Wang, and Jérémy Lhez. Détection d'événements géo-chrono-localisés sur twitter. *Actes du congrès INFORSID, 38e édition*, Éditeurs : Sophie Dupuy-Chessa, Thierry Grison, p. 171–186, Dijon, France, Juin 2020.
2. Hosni Seffih, Myriam Lamolle, Aurélie Pradelles, Ayoub Chouchou, and Christian Fluhr. How ordinary city shift to smart city ? a case study "citizen-centered". *In International Conference Connected Smart Cities 2019*, p. 395–399, Juillet 2019.
3. Hosni Seffih, Myriam Lamolle, Aurélie Pradelles, Christian Fluhr, and Ayoub Chouchou. A new smart citizen-centered transportation application without gps. *In proceedings of the 11th International Conference on Management of Digital EcoSystems*, p. 187–190, 2019.
4. Gaël Guibon, Liana Ermakova, Hosni Seffih, Anton Firsov, and Guillaume Le Noé-Bienvenu. Multilingual Fake News Detection with Satire. *In proceedings of International Conference on Computational Linguistics and Intelligent Text Processing (CICLing), La Rochelle, France*, April 2019.
5. Houda Saadane, Hosni Seffih, Christian Fluhr, Khalid Choukri, and Nasredine Semmar. Automatic identification of maghreb dialects using a dictionary-based approach. *In proceedings of Eleventh International Conference on Language Resources and Evaluation (LREC)*, 2018.
6. Houda Saadane, Damien Nouvel, Hosni Seffih, and Christian Fluhr. Une approche linguistique pour la détection des dialectes arabes (a linguistic approach for the detectionof arabic dialects). *In Actes des 24ème Conférence sur le Traitement Automatique des Langues Naturelles.* Volume 2, p. 242–250, 2017.

# Sommaire

**Introduction générale** **1**

## I Contexte de l'étude 9

### 1 Positionnement de cette recherche 11
1.1 La recherche chez GEOLSemantics . . . 12
1.2 Repositionnement de la problématique . . . 29
1.3 Champs d'expérimentation . . . 32
1.4 Mise en perspective du projet *SafeCity* . . . 33
1.5 Premières conclusions . . . 36

### 2 État de l'art 39
2.1 Analyse des réseaux sociaux . . . 39
2.2 Extraction d'événements et localisation . . . 43
2.3 Fouille de données . . . 46
2.4 Fouille de textes . . . 53

## II Traitement sémantique des réseaux sociaux de type tweets 59

### 3 Désambiguïsation de termes 61
3.1 Notions liminaires . . . 62
3.2 Méthodes de désambiguisation . . . 65
3.3 Solutions de désambiguïsation proposées . . . 67
3.4 Conclusion . . . 79

### 4 Corpus d'alerte 81
4.1 Méthodes de constitution du corpus . . . 82
4.2 Corpus de recherche Twitter . . . 88
4.3 Interfaces de constitution du corpus . . . 90
4.4 Exemple de création de corpus . . . 97
4.5 Corpus d'expérimentation . . . 102
4.6 Premières constatations . . . 105
4.7 Conclusion . . . 108

**III Identification d'événement** **117**

**5 Ontologie événementielle** **119**
5.1 Modèle ontologique . . . . . 120
5.2 Ontologie GEOLSemantics des personnes . . . . . 123
5.3 Ontologies gérant des événements . . . . . 126
5.4 Détection d'événements dans les tweets . . . . . 127
5.5 Représentation des événements dans l'ontologie GEOLSemantics . . . . . 133
5.6 Conclusion . . . . . 146

**6 Amélioration de la localisation des événements** **147**
6.1 Nouvelles règles linguistiques pour GEOLsig . . . . . 149
6.2 Bases de données géographique . . . . . 158
6.3 Première expérimentation . . . . . 179
6.4 Conclusion . . . . . 181

**IV Interprétation de groupes de messages et suivi d'événements** **183**

**7 Suivi des messages** **185**
7.1 Traitement contextualisé des lieux . . . . . 185
7.2 Émission d'alerte . . . . . 188
7.3 Adaptation de la chaîne de traitement GEOLSemantics . . . . . 189
7.4 Expérimentations . . . . . 203
7.5 Conclusion . . . . . 210

**Conclusion** **212**

**Annexe** **219**

# Liste des figures


1 Exemple de rédaction de tweet . . . 3
2 Exemple de code switching français/anglais/arabe dialectal . . . 4

1.1 Schéma simplifié du traitement complet de GEOLSemantics . . . 12
1.2 Exemple de phrases à analyser dans le cas d'un *code-switching* . . . 17
1.3 Résultat du découpage pour le texte de la figure 1.2 . . . 17
1.4 Résultat du découpage pour le texte de la figure 2 . . . 18
1.5 Processus de traitement de l'analyse linguistique profonde . . . 19
1.6 Exemple de représentation d'un arbre syntaxique . . . 24
1.7 L'information vue selon son volume et sa densité (source : [Delort, 2018]) . 31

2.1 Exemple d'une faute orthographique . . . 42
2.2 Exemple d'une faute syntaxique . . . 42
2.3 Exemple d'une faute grammaticale . . . 42
2.4 Exemple de déclaration d'un lieu correcte . . . 46
2.5 Exemple de non déclaration de lieu . . . 47
2.6 Exemple de déclaration d'un lieu imaginaire . . . 48
2.7 Sous-ensemble de l'I.A. . . . 49
2.8 Schéma d'un réseau de neurones à une couche . . . 51
2.9 Représentation *word2vec* . . . 55
2.10 Modèle de langue unidirectionnel . . . 55
2.11 Modèle de langue bidirectionnel . . . 56
2.12 Trois méthodes générales pour affiner BERT [Sun *et al.*, 2019] . . . 57

4.1 Interface : lancement d'une requête . . . 93
4.2 Interface : résultat de la requête . . . 94
4.3 Exemple d'un tweet d'un accident (métaphorique) . . . 101
4.4 Nombre d'utilisateurs de Twitter actifs dans le monde de 2010 à 2018 . . . 106

5.1 Représentation partielle des entités . . . 124
5.2 Hiérarchie partielle des concepts pour les événements . . . 125
5.3 Processus complet de détection d'événements géo-chrono-localisés . . . 132
5.4 Représentation d'un lieu fixe . . . 138
5.5 Représentation d'un lieu flou . . . 139
5.6 Représentation d'un lieu flou avec une interface d'un outil tier . . . 140

6.1 Format XML d'un fichier OSM . . . 161

6.2 Exemple d'une représentation way, node et relation dans OSM . . . . . 162
6.3 Exemple d'une représentation way . . . . . 163
6.4 Interface de géolocalisation des lieux . . . . . 177
6.5 Retweet automatique d'un événement par geolBOT . . . . . 178

7.1 Type de déclaration de lieux . . . . . 187
7.2 Processus d'écriture de *Poire* et ses déclinaisons dans le dictionnaire . . . . 190
7.3 Tweet avec émoticônes . . . . . 191
7.4 Organisation du traitement POSTgal . . . . . 198
7.5 Premier tweet lié à l'événement « acte de violence » autour du marché de Noël . . . . . 202
7.6 Deuxième tweet lié à l'événement « acte de violence » apportant une précision sur le lieu de la figure 7.5 . . . . . 203
7.7 Ambiguïté du terme « Le Président » pour la ville de Nice . . . . . 205

# Liste des tableaux


1.1 Ontologie partielle des événements . . . 27
1.2 Ontologie partielle pour les entités . . . 28

3.1 Relation entre les lieux . . . 74

4.1 Corpus : requête de récupération des tweets pour chaque événement . . . 99
4.2 Récupération des tweets en utilisant des mots-clés . . . 109
4.3 Récupération des tweets par expressions-clés . . . 110
4.4 Création des requêtes avec contexte . . . 111
4.5 Nombre de tweets obtenus par type d'événements pour *Nice*. . . . 112
4.6 Moyenne des caractères par tweet et par type d'événements pour Nice (bounding box) . . . 112
4.7 Utilisation des mentions et de # par type d'événements pour Nice (bounding box) . . . 112
4.8 Pourcentage de vrais événements dans les tweets annotés pour Nice . . . 113
4.9 Nombre de tweets obtenus par type d'événements sur une *bouding box* de *Nice* . . . 113
4.10 Moyenne des caractères par tweet et par type d'événements pour Nice (requête) . . . 113
4.11 Utilisation des mentions et de # par type d'événements pour Nice (requête) 114
4.12 Pourcentage des tweets événement/le nombre de tweets annotés . . . 114
4.13 Nombre de mention des autorités par événements . . . 115
4.14 Exemple de tweets pour chaque événement . . . 116

5.1 Liste des événements selon la durée . . . 135
5.2 Liste des événements par ordre de priorité . . . 136
5.3 Propriétés des événements à étudier . . . 146

6.1 Résultats de comparaison pour l'extraction des lieux avec et sans GEOLsig 155
6.2 Statistiques sur les résultats de comparaison pour l'extraction des lieux avec et sans GEOLsig . . . 155
6.3 Comparaison entre Solr et Elasticsearch . . . 168
6.4 Tests de performance des systèmes GEOLSemantics et BERT . . . 181

7.1 Différences entre BERT et GEOLsig lors de l'expérimentation . . . 206
7.2 Création du corpus d'expérimentation . . . 207
7.3 Résultat de l'expérimentation . . . 208

7.4 Répartition des VP, VN, FP, FN selon les cas de figure . . . . . 211

# Introduction

Le monde actuel est devenu le monde de l'échange d'information à tout va. Nous sommes noyés d'informations pertinentes ou non, dont la véracité est parfois difficile à déterminer et dont la lecture n'est pas toujours facile. Quand il s'agit de détecter des événements à partir de textes qui, *a priori*, sont bien structurés, on peut penser que nos technologie de pointe en sont capables. Malheureusement, il n'en est rien. En effet, les sources de données textuelles sont plus diversifiées par les formes de communication à disposition (sms, mms, courriel, fichiers pdf ou autres, etc.), mais aussi dans leur mode de rédaction.

Dans le contexte des réseaux sociaux, les informations sont souvent écrites dans un style particulier en modifiant les caractéristiques orthographiques et/ou grammaticales afin de réduire la longueur du texte car le nombre de caractères autorisé est limité, avec l'utilisation en outre de mots-dièses (dits *hashtags*) pour mettre en exergue un thème particulier. Tout ceci augmente la difficulté de détection d'événements. La détection et le suivi d'événements à partir de textes analysés en temps réel ou semi-réel reste donc un défi.

La détection d'événements « géo-chrono-localisés » n'est pas chose aisée sur Twitter sachant qu'elle doit se faire au plus tôt donc en un minimum de tweets [Ying *et al.*, 2018]. De façon générale, beaucoup d'ambiguïtés peuvent survenir des termes utilisés s'ils ne sont pas replacés dans leur contexte. Par exemple, le terme « Nice » peut faire référence à la ville du sud de la France mais aussi à la notion de « joli » (selon l'anglais) alors que la conversation est en français. Autant nous sommes capables intuitivement de faire cette distinction et donc de localiser ou non l'événement, autant cela devient difficile pour une machine ou un programme.

Dans le cas particulier des réseaux sociaux, notamment dans les tweets, le vocabulaire

et la tournure des phrases sont encore plus succincts puisque contraints à 144 caractères jusqu'en 2017 [1] [Perez, 2017]. Par exemple, il apparaît beaucoup plus de mots contractés ( « tkt » pour « Ne t'inquiètes pas »), de phonétisations (« foto » pour « photo » ou « C » pour « c'est »), de pictogrammes, etc. Quant à la détection d'événements, nous pouvons être guidés par les *hashtags* mais cela ne nous permettra pas de déterminer où et quand se déroule(-era) l'événement. Ces informations sont essentielles lors du déclenchement d'alertes automatiques pour des risques climatiques, des mises en sécurité de personnes, etc., en temps réel ou semi-réel.

Ce travail de recherche s'inscrit dans cette perspective pour, d'une part, extraire et améliorer le plus possible la compréhension de la sémantique d'un tweet, et d'autre part, augmenter la pertinence des alertes en associant aux événements une temporalité et une localisation acceptables.

Cette thèse entre dans le cadre d'une convention CIFRE entre GEOLSemantics et l'Université Paris 8. GEOLSemantics est un éditeur de logiciels innovants dans le monde des technologies de l'information et de la communication et particulièrement dans les domaines de la linguistique et de la sémantique, créé en 2009 [2].

En nous fondant sur les outils d'extraction de connaissances développés par GEOLSemantics, capables d'extraire des informations structurées sur des entités nommées ou des objets, notre premier objectif va concerner la définition de règles de désambiguïsation afin de déterminer le plus précisément possible les lieux et dates des événements ; c'est-à-dire des règles qui, d'après les échanges de tweets, vont mettre en évidence un événement nécessitant le déclenchement d'une alerte.

Notre deuxième objectif pour affiner la pertinence du déclenchement d'une alerte va être de compléter une ontologie existante dont la modélisation va être dirigée par un corpus *ad hoc* au contexte des événements afin d'inférer des patrons (ou *patterns*) « cause à effet » entre certains événements qui ont ce type de rapport. À l'heure actuelle, l'ontologie qui sert de référence à cette extraction est orientée sécurité et couvre un grand nombre d'événements et d'attributs. Bien entendu, cette ontologie est modifiable pour rechercher d'autres types

1. passage à 280 caractères en novembre 2017
2. voir `http://www.GEOLSemantics.com/index.php/fr/societe/`

d'événements ou pour étendre les types d'événements reconnus.

Les outils d'extraction de GEOLSemantics sont fondés sur une analyse morphosyntaxique profonde généraliste puis sur une extraction sémantique relevant les concepts décrits dans l'ontologie de l'application grâce à des règles d'extraction qui utilisent les résultats de l'analyse morphosyntaxique profonde. Bien que le traitement soit robuste à la présence de quelques fautes, il doit être adapté pour prendre en compte les rédactions particulières visibles sur les réseaux sociaux comme les tweets. Les textes apparaissent avec une écriture plus phonétique qu'orthographique avec des mots collés ou simplifiés. De même, la syntaxe est à la fois incorrecte (mots manquants) mais plus simple (l'ordre des mots est respecté) du fait du faible nombre de caractères (cf. figure 1).




FIGURE 1 – Exemple de rédaction de tweet

Aussi, ceci nécessite une adaptation profonde des traitements pour obtenir de la morpho-syntaxe les différentes interprétations que les hypothèses de reconnaissance de mots vont faire apparaître avec une désambiguïsation syntaxique limitée. La sémantique apportée par l'ontologie doit permettre de choisir l'interprétation correcte dans le cas où le message fait référence à un événement prévu dans l'ontologie.

Le domaine d'application est l'extraction d'événements qui demandent l'intervention des autorités locales. Les événements doivent être situés dans le temps et dans l'espace afin d'être assez précis pour pouvoir être vérifiés, ou pour permettre une intervention des autorités. Notons que nous limitons cette étude à des événements présents ou du passé

très récent et aux événements prévus (futurs), c'est-à-dire par exemple des incendies et des inondations sur la ville de Nice, ou l'organisation de rassemblement de personnes toujours à Nice, avec un corpus *ad hoc* en respect avec la RGPD[3].

Pour interpréter les événements, un message unique peut ne pas être suffisant. De ce fait, les événements seront identifiés en s'appuyant sur un ensemble de messages y compris leurs métadonnées (par exemple, pseudo de l'émetteur, pseudo de celui à qui il répond, date, heure, éventuellement lieu, etc.). Le contexte d'utilisation étant strict, il est à noter que celles-ci ne seront pas stockées, mais traitées à la volée, dans le respect de la RGPD.

Le travail a été réalisé dans un premier temps sur le français, puis élargi à l'arabe *"Modern Standard Arabic"* (MSA) et l'arabe dialectal en se fondant sur les travaux réalisés pendant le projet *Orelo* [Saadane *et al.*, 2018].

Il existe beaucoup d'alternances codiques (ou *code-switching*) dans les tweets en français. Cela se produit lorsque l'auteur d'un tweet alterne entre deux ou plusieurs langues ou variétés de langues dans le contexte d'un seul tweet. Dans le cas de tweets en français, il s'agit souvent de l'utilisation de mots anglais ou arabes au sein même du texte (voir l'exemple de la figure 2).




FIGURE 2 – Exemple de code switching français/anglais/arabe dialectal

Saâdane [2015] a répertorié les techniques scripturaires utilisées par les rédacteurs arabes qui font du *code-switching* avec du français. Ce travail a permis à GEOLSemantics de développer un système qui permet de :

— segmenter le message selon les langues présentes ;

3. voir `https://www.cnil.fr/fr/comprendre-le-rgpd`

— traduire vers une seule langue afin d'unifier la langue du message,

— traiter le message.

Nous détaillerons cette technique dans le chapitre 1.1.1.

# Problématique

La détection et le suivi d'événements à partir de textes analysés en temps réel reste un défi. Les sources de données textuelles sont diverses en particulier dans leur mode de rédaction. Les réseaux sociaux véhiculent souvent des informations écrites de manière simplifiée ou écrites par des utilisateurs qui ne maîtrisent pas l'orthographe ou qui, par paresse, simplifient l'écriture.

Nous allons maintenant passer en revue les différents verrous scientifiques actuels dans ce domaine et en rapport avec ce travail de recherche.

## Constitution d'un corpus sur un territoire particulier

Pour les besoins du projet *SafeCity* (détaillé dans la section 1.4), nous devons construire un corpus sur une liste finie d'événements avec comme contrainte forte que l'observation doit se faire sur des territoires déterminés, en l'occurrence pour nos expérimentations, Nice et la Défense. Ceci implique une problématique particulière pour essayer de trouver le maximum de tweets qui sont émis du territoire concerné et qui sont liés aux événements attendus.

Il y a donc toute une étude faite appuyée par des tests pour savoir quel pourcentage donne des coordonnées fiables. Comme il peut y en avoir très peu, il faut procéder avec des requêtes ou filtres complexes mariant recherche sur les textes, *hashtags* et métadonnées.

Deux approches sont envisageables : soit optimiser la détermination du lieu et ensuite vérifier l'événement, soit réaliser une optimisation sur l'événement et ensuite vérifier le lieu. Il est possible de mixer les deux approches sans perdre trop de messages. Le but est d'éviter le maximum de messages non pertinents en perdant le moins possible de

messages pertinents ; en d'autres mots, éviter l'extraction de faux positifs (fausses alertes, métaphores, ...) tout en maximisant la précision du résultat.

## Traitement sémantique des réseaux sociaux

Même si l'orthographe liée à l'écriture des messages s'est bien améliorée depuis les premiers textos, il reste que ces messages sont souvent écrits phonétiquement, avec des mots collés et/ou des abréviations (en particulier, quand il y a une limite de 280 caractères). Les outils d'analyse linguistique et d'extraction sémantique tels que ceux développés par GEOLSemantics doivent être adaptés pour être robustes à ce type de rédaction. Cela se traduit par des interprétations de mots plus ambiguës et par une syntaxe plus simple avec un manque de mots outils mais, par contre, avec une construction simplifiée des phrases.

La syntaxe ne pouvant éliminer toutes les interprétations fausses, c'est la sémantique qui est représentée par l'ontologie décrivant le besoin utilisateur qui va servir de filtre. On ne cherche pas ici à comprendre tous les messages mais bien à interpréter ceux qui contiennent des informations sur des événements qui intéressent les utilisateurs finaux.

## Identification d'événements

L'extraction d'événements consiste à : (i) identifier des événements et les informations les concernant recherchés par un utilisateur dans un texte donné ; (ii) structurer ces informations identifiées.

Il s'agit ici d'identifier les événements que veut repérer l'utilisateur en les caractérisant par au minimum un lieu et une date sachant que l'on s'intéresse dans ce projet aux présent ou passé immédiat pour lancer des alertes et au futur pour prévoir de possibles événements.

La principale différence entre les approches proposées par la littérature et celle de GEOLSemantics porte sur le fait que GEOLSemantics ne s'appuie pas uniquement sur des technologies du traitement du langage naturel mais aussi sur l'utilisation de technologies du web sémantique (ontologie, raisonnement) pour pallier le manque de rigueur dans la rédaction des messages.

## Suivi d'événements

Il semble évident que repérer un événement dans un seul message n'est pas suffisant à plusieurs titres notamment par rapport à sa temporalité, sa localisation, etc.. D'autre part, un tweet ne peut pas toujours refléter à lui seul l'importance d'un événement. Il faut être capable d'identifier qu'un autre message relate le même événement ou non citeJackoway2011. C'est indispensable pour ne pas « noyer » l'alerte par des informations redondantes. Il faut aussi être capable de compléter l'information au fur et à mesure que de nouveaux messages arrivent. Enfin, en ce qui concerne des événements annoncés, il faut pouvoir les suivre pour confirmer s'ils arrivent ou non, s'ils auront lieu ou non.

## Organisation du manuscrit

Ce mémoire de thèse retrace mes trois années de recherche sur la détection d'événements géo-chrono-localisés dans des tweets et, ce, le plus rapidement possible afin de lancer des alertes pertinentes auprès d'organismes publics.

Il est articulé comme suit. Après une brève introduction, la première partie présente le contexte de l'étude. En effet, cette étude a été réalisée au sein de la société GEOLSemantics dans le cadre d'une thèse CIFRE avec le LIASD [4]. Le positionnement de la problématique est important dans ce contexte car il est dépendant de plusieurs facteurs tels que la constitution d'un corpus adéquat (cela a nécessité plus de travail que prévu), le fait que le ratio entre information pertinente et nombre de tweets collectés nous a placé dans un contexte de *big data* ou des notions de confidentialité. Par exempke, comme nous l'aborderons dans le chapitre 4, nous nous sommes positionné par rapport à la RGPD [5], le règlement général sur la protection des données, qui est entrée en vigueur en mai 2016 et appliquée à partir de mai 2018 donc au cours de cette thèse.

Les champs d'expérimentation sous-jacents sont en lien étroit avec les projets de recherche de l'entreprise comme le projet *SafeCity* qui est exposé dans le chapitre 1.4.

4. Laboratoire d'informatique Avancée de Saint Denis, Université Paris 8
5. `https://www.cnil.fr/fr/comprendre-le-rgpd`

La deuxième partie aborde le traitement sémantique des réseaux sociaux de type tweets. Pour ce faire, un état de l'art fait le point sur les recherche actuelles d'analyse des réseaux sociaux, l'extraction d'événements et la fouille de données notamment textuelles. Le chapitre 3 présente les techniques de désambiguïsation de termes et lors du chapitre 4, nous spécifions le corpus d'alerte.

La troisième partie se focalise sur l'identification d'événement par la présentation de l'ontologie événementielle (cf. chapitre 5), l'amélioration de la localisation des événements grâce à la mise en place de règles linguistiques et l'utilisation d'une base de données géographiques *ad hoc*.

La dernière partie de ce travail concerne l'interprétation de groupes de messages et le suivi d'événements pour l'émission d'alerte. Les résultats de nos expérimentations sont présentés montrant le bien fondé de notre approche mais aussi ses limites.

Nous terminerons, lors du chapitre 7.5, par un bilan général et quelques pistes de recherche sur les événements géo-chrono-localisés à solutionner dans l'avenir. ‘

# Première partie

# Contexte de l'étude

# Chapitre 1

# Positionnement de cette recherche

## Sommaire


**1.1 La recherche chez GEOLSemantics . . . . . 12**

**1.2 Repositionnement de la problématique . . . . . 29**

**1.3 Champs d'expérimentation . . . . . 32**

**1.4 Mise en perspective du projet *SafeCity* . . . . . 33**

**1.5 Premières conclusions . . . . . 36**


Dans ce chapitre, nous allons délimiter le contexte général de l'étude en présentant tout d'abord l'environnement de la recherche au sein de l'entreprise GEOLSemantics, entreprise spécialisée dans la recherche d'information documentaire. Pour ce faire, elle a mis en place une architecture modulable afin de paramétrer le processus des différents traitements linguistiques selon les documents à gérer. Nous ferons le point sur les modules développés pour une gestion générale de documents qui ont constitué le socle de nos travaux de recherche ; et nous mettrons en exergue les limites de ces derniers dans le cas du traitement de messages courts ayant une syntaxe et un vocabulaire approximatif comme peuvent l'être des tweets.

## 1.1 La recherche chez GEOLSemantics

GEOLSemantics est un éditeur de logiciels linguistiques et sémantiques structurant les informations contenues dans des textes.

L'entreprise se concentre principalement sur le marché de la sécurité en France. Les traitements se font sur des textes français et/ou arabes mais peuvent s'étendre à des textes en anglais, russe ou chinois. Elle travaille aussi sur des textes juridiques, sur la réglementation des données personnelles RGPD [1] et sur la détection d'événements à partir des réseaux sociaux.1.1

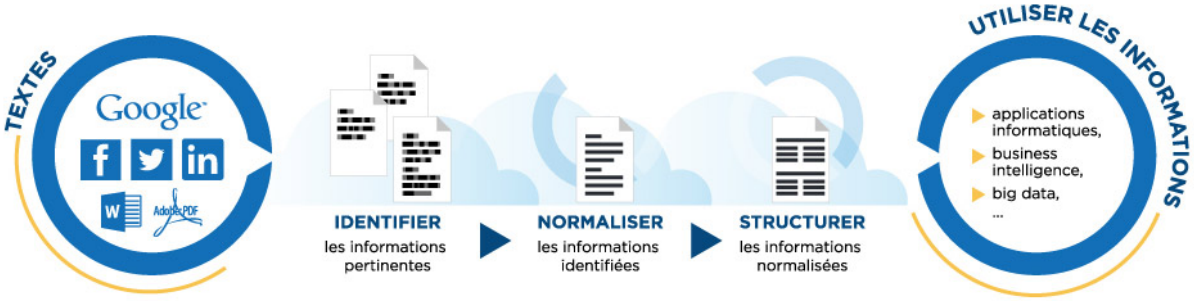



FIGURE 1.1 – Schéma simplifié du traitement complet de GEOLSemantics

Pour ce faire, GEOLSemantics possède déjà un système "de bout en bout" d'extraction de l'information, s'appuyant sur une ontologie et des règles linguistiques d'extraction. Les données textuelles en entrée sont transformées en document XML contenant pour chaque élément du texte originel son lemme (verbes à l'infinitif, adjectifs au masculin singulier, etc.), sa catégorie linguistique (nom, verbe, etc.), ses propriétés syntaxiques et sémantiques (féminin pluriel, organisation, etc.) et ses relations avec les autres éléments du texte, ainsi que les entités nommées typées qu'il contient. Le texte ainsi représenté peut, à la demande, être ensuite traité par notre moteur sémantique qui, au-delà de la normalisation, va extraire des concepts et leurs propriétés, quelle que soit la formulation utilisée, pour donner un résultat en RDF en conformité avec l'ontologie définie.

Afin d'assurer le processus complet de traitement depuis un texte brut passé dans les analyseurs GEOLSemantics jusqu'à une version sémantique en RDF porteuse de connaissances, il y a une transformation intermédiaire en XML représentant l'analyse linguistique

1. Règlement Général sur la Protection des Données

(Texte=>XML=>RDF). Le passage par le XML permet de normaliser le texte (lemmes, entités nommées, relations) pour enchaîner avec une analyse sémantique facilitée par cette représentation. Ce découpage en différents modules (analyse linguistique, analyse sémantique) permet une plus grande flexibilité dans la recherche d'information à des degrés divers.

L'intérêt de ce processus complet est de pouvoir paramétrer et configurer un outil allant de la création d'un moteur de recherche fondé sur des mots-clés sur un corpus spécifique (juridique, scientifique, etc.) à la détection d'informations pertinentes sur de gros corpus pour tout utilisateur final. Il est à noter que jusqu'à présent le texte brut devait respecter les règles orthographiques et linguistiques propres à la langue de sa rédaction (hors langage SMS).

Non seulement les modules de ce processus peuvent changer de configuration facilement selon les besoins de l'utilisateur final ; mais il est aussi facile d'adapter les ressources selon un domaine spécifique.

Nous allons détailler ces modules dans la section (1.1.1).

### 1.1.1 Traitement linguistique de documents

Différents modules-clé interviennent lors du traitement sémantique de documents textuels mais un module est plus au cœur du processus à savoir *HTFST* qui est remplacé progressivement par *GLAS* une forme plus évoluée de moteur linguistique :

1. HTFST : cet outil se base sur la technologie générique des transducteurs à états finis, afin de lemmatiser et tokeniser des textes, extraire les entités nommées ainsi que les relations. Malheureusement, HTFST est lent car il n'a pas été optimisé pour le Traitement Automatique des Langues (TAL), et pour parvenir à une analyse linguistique profonde, il crée un très grand automate qui va reproduire tous les choix possibles sans que nous puissions améliorer les algorithmes de parcours et de décision des meilleurs chemins.
2. GLAS : il s'agit d'un outil conçu pour le français, l'arabe et qui est en train d'être

étendu à l'anglais. Il a été développé en interne, après une étude des différents outils OpenSource disponibles afin de répondre parfaitement aux problématiques du TAL que GEOLSemantics rencontre dans ses traitements à savoir le plurilinguisme et la diversité des moyens d'expression entre autres. À terme, il remplacera HTFST.

Selon la figure (1.1), GLAS est chargé de superviser tous les modules intervenant dans le processus de traitement jusqu'à l'étape *Structurer* incluse.

Les règles linguistiques sont celles utilisées par l'ancien système avec une nouvelle syntaxe pour le français et l'arabe. Elles sont en cours de réécriture pour l'anglais ; ce qui rendra GLAS complètement opérationnel. De plus, les moteurs d'exécution ont été réécrits de manière optimisée pour outrepasser la lenteur d'HTFST et assurer une plus grande souplesse du système afin de l'adapter aux moyens de communication comme les réseaux sociaux.

Pour permettre une comparaison des résultats et des temps de traitement obtenus entre l'ancienne et la nouvelle version, un système d'analyse de non régression a été développé.

GLAS permet le bon fonctionnement du module GAL (pour *GEOLSemantics Analyse Linguistique* que nous détaillerons dans le chapitre 7) qui est le premier module exécuté dans le processus des traitements lingistiques de GEOLSemantics. Ce dernier effectue l'analyse linguistique des textes. Il extrait les lemmes, leur attribue des catégories grammaticales et des propriétés morpho-syntaxiques. Il détermine aussi les relations entre lemmes. Enfin, il reconnaît et type les entités nommées (EN).

Ensuite, le module GEX (pour *GEOLSemantics EXtraction*) et le module GEXIH (*GEOLSemantics Extraction Interlingual Homogeneisation*, que nous détaillerons dans le chapitre 7) se chargent de l'analyse sémantique. GEX effectue une extraction d'information au niveau de la phrase en se fondant sur une analyse sémantique guidée par une ontologie ; tandis que GEXIH réalise une homogénéisation et une désambiguïsation des informations au niveau des documents. GEXIH se charge aussi de la désambiguïsation temporelle pour les dates relatives et de la mise en relation des informations extraites avec d'autres bases de connaissances (DBpedia, etc.). Il permet aussi de fusionner les informations issues de différents documents. Par exemple, aujourd'hui sera transformé en format AAAAMMJJ-

Thhmmss selon la date indiquée dans les métadonnées du document ou celle présente dans le texte lui-même. Ce format correspond à :

— AAAA : année,

— MM : mois,

— JJ : jour,

— T : séparateur entre la date et l'heure,

— hh : heure,

— mm : minute,

— ss : seconde.

Abordons maintenant en détail le fonctionnement de chaque module.

### Analyseur linguistique GAL

L'analyseur linguistique de GEOLSemantics (GAL) fait une analyse morpho-syntaxique automatique transformant du texte brut en une version textuelle structurée XML. Il est alors plus facile d'appliquer des traitements d'extraction de connaissances, de recherche de documents multilingues et de reformulation multilingue à partir de cet arbre XML. Chaque traitement va bénéficier d'une analyse linguistique afin de faciliter le déclenchement de règles propres d'extraction, de reformulation et de recherche d'information, notamment pour répondre à des exigences particulières tant du point de vue de l'utilisateur que du point de vue du formatage du texte comme nous allons le voir dans la section (7.3.2).

Le format d'entrée du module GAL est donc un texte brut. Le résultat de l'analyse est au format XML validé par un fichier DTD prédéfini. La racine du fichier est constituée par la balise `<text>... <\text>` représentant le texte, et qui comprend à son tour une à plusieurs balises `<par>...<\par>` indiquant les paragraphes du texte. La balise primordiale est celle qui sert à segmenter les paragraphes en phrases, notée `<sent>...<sent>` ( pour "*sentence*" en anglais). Elle contient les balises plus spécifiques :

1. `<word>...<\word>` dont chacune entoure un mot et ses informations morphosyntaxiques ;

2. `<en>...<\en>` qui est le marqueur pour une entité nommée ;
3. `<rel>...<\rel>` (pour "*relation*") qui marque une relation syntaxique.

L'analyseur est conçu pour traiter des langues très diversifiées (anglais, français, arabe, chinois) quel que soit le type de texte même si le document est rédigé en plusieurs langues (grâce au *code-switching*). Le *code-switching* concerne un changement de langue d'un paragraphe à un autre, d'une phrase à une autre mais aussi à l'intérieur d'une même phrase.

Dans ces cas-là, le texte brut est découpé en différents paragraphes, puis en phrases en prenant en compte la ponctuation et les sauts de lignes. En l'absence de saut de ligne, on considérera qu'un paragraphe est limité à 399 caractères pour éviter de saturer la mémoire de la machine. La contrainte de la mémoire sera résolue, le découpage ne sera plus limité à 399 caractères.

La phrase, quant à elle, peut aussi être découpée dans le cas du *code-switching*. Elle est alors séparée en segments selon la langue. Si les segments sont supérieurs à trois mots, ils seront analysés par le module GAL en utilisant, pour chaque segment, les ressources de sa langue respective ; sinon le segment est traduit vers la langue utilisée majoritairement dans le texte brut pour un traitement avec les ressources de cette dernière (la langue par défaut étant le français) [Saâdane, 2015].

Par exemple, le programme de segmentation dans le cas du *code-switching* vers l'arabizi (arabe translittéré en caractères latins) effectue les opérations suivantes :

— utiliser les analyseurs du français et de l'anglais de GEOLSemantics pour annoter les mots écrits en caractères latins et identifier si les mots sont écrits en français, anglais ou en arabe latinisé (arabizi) ;

— utiliser les analyseurs morphologiques du MSA [2] et ceux de l'arabe dialectal pour annoter les mots écrits en arabe ;

— déterminer et identifier la langue des segments ;

— valider les frontières pour chaque segment, comme sur la figure 1.3.

On trouve le même phénomène dans la twittosphère française, la figure 1.4 représente le découpage du tweet 2 présenté comme exemple dans l'introduction.

2. Modern Standard Arabic

```
<doc docid="Yakhoo_comment3" articleURL="http://z4.invisionfree.com/Yakhoo/ar/t73.htm"
author="3assimi" date="October 4, 2007" time="11:32" subtitle="M3askri ra7 yekhtob
(Yakhoo)">
<comment> hahahaha .. et ca me rappelle une autre... wahed kal el yemah zawjini, mais choufili
wahda chaba, c tout ce que je demande.. yemah kaletlou ya wlidi ezine 3omrou ma bna dar ... alors
kalelha .. choufili maçon alors :P
</comment>
</doc>
```

FIGURE 1.2 – Exemple de phrases à analyser dans le cas d'un *code-switching*

```
<doc docid="Yakhoo_comment3" articleURL="http://z4.invisionfree.com/Yakhoo/ar/t73.htm"
author="3assimi" date="October 4, 2007" time="11:32" subtitle="M3askri ra7 yekhtob (Yakhoo)"
lang="DZ">
<comment>
<French> hahahaha .. et ca me rappelle une autre... </French>
<Arabizi> wahed kal el yemah zawjini, mais choufili wahda chaba, </Arabizi>
<French> c tout ce que je demande.. </French>
<Arabizi> yemah kaletlou ya wlidi ezine 3omrou ma bna dar ... alors kalelha .. choufili maçon
alors :P </Arabizi>
</comment>
</doc>
```

FIGURE 1.3 – Résultat du découpage pour le texte de la figure 1.2

Beaucoup de travaux ont été réalisés sur le *code-switching* notamment Habash [2008] qui a présenté un guide pour l'identification du contenu dialectal dans un contenu arabe en accordant une grande attention à ce phénomène. Ce travail offre un corpus annoté de 59 documents contenant 19 000 mots environ.

Sauf cas particulier de *code-switching*, la phrase sert d'unité d'analyse pour les traitements qui suivent la transformation du module GAL.

En effet, à notre connaissance, selon différents travaux de recherche, une phrase est un groupe de mots (ou peut-être un seul mot), portant souvent une signification idiomatique spéciale. En analyse linguistique, les mots fonctionnent comme des constituants dans la syntaxe d'une phrase [Kroeger, 2019]. La détection de phrases est un enjeu majeur en traitement linguistique. Pour les langues occidentales, la problématique principale est de distinguer la sémantique liée au point qui peut avoir plusieurs rôles : (i) soit c'est un indicateur de fin de phrase, (ii) soit il a d'autres fonctions comme la fin d'abréviations (par exemple, *etc.* ) ou le séparateur de nombres décimaux [Hung *et al.*, 2010]. GEOLSemantics

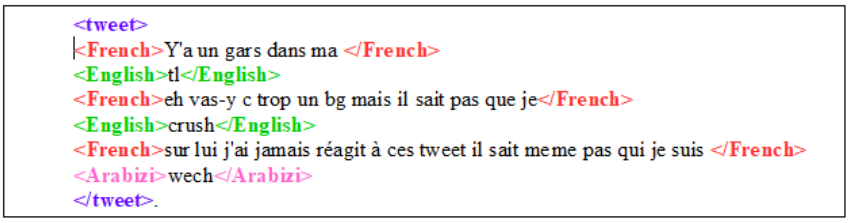



FIGURE 1.4 – Résultat du découpage pour le texte de la figure 2

se fonde sur un automate à états finis qui effectue à la fois le découpage en "mots" (ou tokens) et le découpage en phrases. Lorsque l'automate est dans l'état "point", il utilise une expression régulière pour déterminer dans quel contexte il se trouve : abréviation, sigle, points de suspension, points décimaux, points de fin de phrase, points dans une adresse mail ou dans l'adresse d'un site web. Tous les caractères ambigus sont traités de la même manière, comme la virgule (qui peut aussi appartenir à un nombre).

Une fois découpée, chaque phrase subira les traitements suivants comme le montre la figure 1.5 :

- la tokenisation,
- la catégorisation,
- la lemmatisation,
- la désambiguïsation concernant
  - la reconnaissance des entités nommées,
  - l'identification des relations syntaxiques,
  - la gestion de la négation, des modalités et des pronoms.

Une fois tous les traitements exécutés, le résultat obtenu pour chaque phrase sera concaténé avec le précédent dans l'ordre d'apparition des phrases. Le résultat final global va contenir les résultats de l'analyse morphosyntaxique, de l'analyse syntaxique, de la reconnaissance des entités nommées et de la résolution d'anaphore. Ce résultat final sera ensuite utilisé dans les processus d'indexation de document, de reformulation monolingue ou bilingue, ou d'extraction de connaissances. GAL, en soi, peut être utilisé aussi pour une première phase de traitement des tweets mais les résultats obtenus ne sont pas suffisants quand le but fixé est l'extraction d'événements géo-chrono-localisés au plus tôt. Aussi, une part

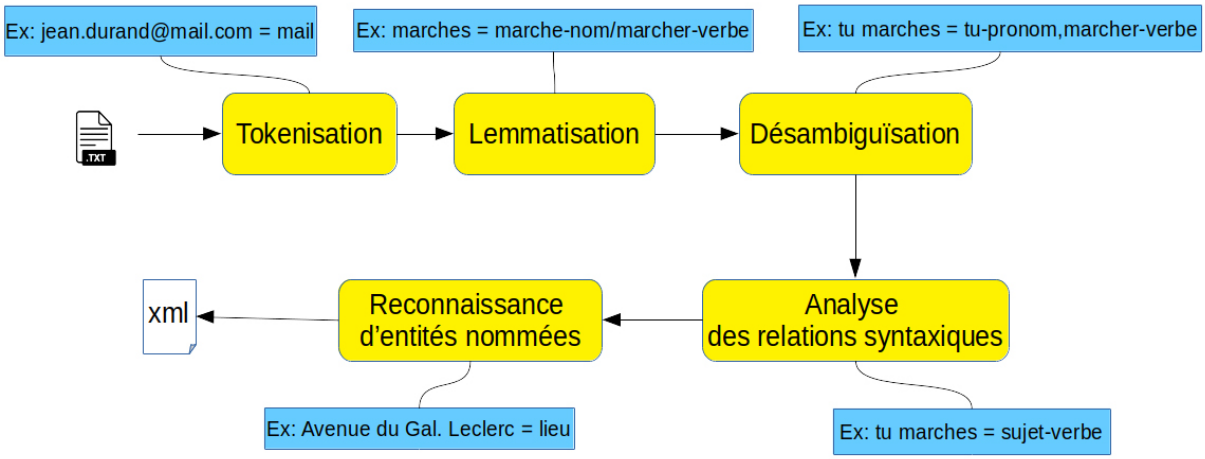



FIGURE 1.5 – Processus de traitement de l'analyse linguistique profonde

importante de ce travail va consister à analyser plus finement la connaissance que nous pourrions extraire du résultat généré par GAL, notamment pour l'extraction des entités nommées, des liens entre elles tout en prenant en compte la temporalité.

**Les composants GAL** Détaillons maintenant les composants analysant une phrase.

**La tokenisation** : le découpage du texte en *tokens* est effectué grâce à un automate à états finis. Un token est une séquence de caractères délimitée par des espaces et/ou des ponctuations. Pour les caractères ambigus, des expressions régulières permettent de décider s'il convient de procéder à un découpage ou non sur celui-ci. Ensuite, à chaque token sont attribuées des classes typographiques ; par exemple *Number* pour "20", *CapitalizedWord* pour "Twitter", "Facebook". Les expressions régulières permettent également de capturer d'autres types de segments tels que les acronymes, les abréviations, les adresses mails, les numéros de téléphone ou encore les balises. Des dictionnaires de langues et des règles de découpage complètent l'utilisation d'expressions régulières pour la résolution de problèmes tels que la concaténation des mots ou l'ajout de suffixe ou de préfixe. La position (de début et de fin) de chaque token est préservée pour pouvoir créer un lien entre notre extraction et le contenu du texte.

**La catégorisation** : après l'étape de tokenisation, il est nécessaire d'attribuer une catégorie à chaque token afin de donner une information syntaxique à chaque mot en se fondant sur des dictionnaires propres à chaque langue. Cette étape nous permet

aussi d'identifier des expressions telles que "pomme de terre", "afin que" grâce à un dictionnaire d'expression.

**La lemmatisation** : c'est une tâche importante qui consiste à retrouver la forme normalisée non fléchie, d'un mot telle que l'infinitif d'un verbe. Cependant les informations sur le temps du verbe, le genre du mot, des modalités, etc., sont conservées.

**La désambiguïsation** intervient sous trois formes à savoir :

— la reconnaissance des entités nommées : la tâche de reconnaissance d'entités nommées est très importante pour le reste des traitements puisqu'elle consiste en l'identification des entités qui seront reliées aux éléments à extraire plus tard tels que les événements. Les entités nommées considérées sont les personnes, les organisations, les lieux, les dates, les mesures et les quantités, et les objets (véhicules, produits, etc.).

En l'absence d'un système de vérification d'entités nommées, les résultats sont filtrés de façon très rigide. En effet, certaines règles d'extraction comme celles des lieux induisent malheureusement beaucoup de faux positifs (*i.e.* des lieux qui ne sont pas réellement des lieux) ou de faux négatifs (*i.e.* des lieux qui sont réellement des lieux mais non reconnus en tant que tel).

Par exemple, soit la phrase *Je suis à laplace.* ; sans majuscule, *laplace* sera annotée par GAL en tant que mot inconnu puisqu'ici la règle d'extraction qui sera appliquée est :

```
Règle i : "à" suivi d'un mot en majuscule
                                => le mot en majuscule est un lieu.
```

Mais en interrogeant une base de données de lieux, *laplace* sera annoté en tant que lieu et non comme mot inconnu, malgré le fait qu'aucune des règles d'extraction n'était supposée l'extraire.

La détection des entités nommées se fait soit en s'aidant (i) de dictionnaires construits à partir des données du *Linked Open Data* (LOD) [3], de Geonames [4] pour les données géographiques et de DBPedia [5] pour les données sur les per-

3. https://lod-cloud.net
4. https://www.geonames.org
5. https://wiki.dbpedia.org

sonnes et les organisations, (ii) des patrons morphosyntaxiques en utilisant des annonceurs ; par exemple, les mots "ville" ou "city" introduisent des lieux, "société" ou "ltd" sont considérés comme étant des annonceurs de compagnies. Certaines entités nommées sont composées de plusieurs éléments qui doivent être regroupés dans cette étape, tels que le patronyme, le prénom, les noms additionnels ou encore le surnom pour les personnes. Dans cette étape, nous réalisons aussi la reconnaissance des nombres, des mesures et des quantités. Les chiffres correspondant aux valeurs numériques sont normalisés et ceux écrits en lettres sont alors transformés en nombre. Une fois les nombres reconnus, vient le traitement des dates. Elles sont normalisées dans le standard ISO-8601 2[6], YYYYMMDDTHHMMSS. Les parties inconnues d'une date sont remplacées par des X. Par exemple, la date "Mars 1988" apparaît sous la forme : 198803XXTXXXXXX.

Mais certaines représentations ne sont pas possibles avec la norme ISO. Aussi, GEOLSemantics gère les intervalles par crochet ouvrant ou fermant - YYYYMMDDTHHMMSS - / - YYYYMMDDTHHMMSS - crochet ouvrant ou fermant sachant que :

— si une date est incluse alors on utilise un crochet tourné vers l'intérieur,

— si une date n'est pas incluse alors on utilise un crochet tourné vers l'extérieur,

— quand une date n'est pas définie dans le passé alors on utilise un crochet fermant,

— quand une date n'est pas définie dans le futur alors on utilise un crochet ouvrant,

— la partie THHMMSS est facultative dans les deux notations (date simple et intervalle).

Voici quelques exemples pour illustrer les différentes représentations possibles :

— jeudi 20 août 2020 : 20200820

— jeudi 20 août 2020 à 11h30 : 20200820T113000

6. voir `https://www.iso.org/standard/70908.html`

- du 1er septembre au 15 septembre 2021 : [20210901/20210915]
- depuis le 5 janvier 2020 : [20200105/XXXXXXXX[
- avant le 5 janvier 2020 : ]XXXXXXXX/20200105]
- après le 20 août 2016 : ]20160820/XXXXXXXX[

— Identification des relations syntaxiques : cette étape permet de mettre en évidence l'arrangement des catégories grammaticales dans une phrase (exemple : sujet-verbe-complément) et reconnaître les relations de dépendance entre les mots. Pour cela, GEOLSemantics utilise une analyse linguistique fondée sur la grammaire définie par Tesnières [Franić, 2011]. Selon Tesnières, les relations syntaxiques entre les mots peuvent être représentées par des relations binaires entre une tête (gouverneur de la relation) et son dépendant. La tête est définie comme le mot qui supporte la relation avec les autres mots de la phrase. Quant au dépendant, il apporte plus de précisions et de détails sur la tête. En parallèle, d'autres mots-outils tels que les prépositions ou encore les déterminants sont considérés comme étant des éléments tiers à la relation pour relier la tête et le dépendant. Le modèle syntaxique de GEOLSemantics est basé sur des patrons syntaxiques créés manuellement qui permettent de représenter le lien entre deux mots : les relations à l'intérieur d'une phrase nominale (notée GN) ; la relation Agent-Verbe (notée AV), la relation Verbe-Complément (notée VC), la relation attribut du sujet (notée ATTRS), et la relation CIRCONSTANT ; la figure 1.6 représente l'arbre syntaxique de l'exemple *"A car bomb killed three people outside a hospital in the eastern Libyan city of Benghazi on Monday."*

— Gestion de la négation, des modalités et des pronoms : le résultat de l'analyse morphosyntaxique est un arbre syntaxique. Elle est suivie par la résolution des pronoms (relier les pronoms à leurs référents), la gestion des modalités et de la négation. Le processus est de sélectionner chaque pronom selon la structure de la règle syntaxique correspondante. Chaque relation possède des informations morphosyntaxiques comme le genre ou le nombre ainsi que des informations sémantiques qui indiquent le type de l'entité telle que personne, lieu ou objet. Ces éléments nous permettent de trouver le bon référent puisque chaque pronom

remplace un genre et un nombre particulier. Une fois le bon candidat sélectionné, le pronom est alors substitué par son référent. Dans l'exemple *« Paul envoie un cadeau à Sarah pour son anniversaire. Elle l'a beaucoup apprécié. »*, les informations morphosyntaxiques et sémantiques indiquent que Sarah est du genre "Féminin" et de type "Personne". De ce fait, le pronom *Elle* se réfère à cette personne. Le *l'* fait référence très certainement à une entité de type "Objet". Remarquons que l'exemple donné est complexe, car selon la sémantique du verbe *l'* peut être interprété de manière différente. Si la deuxième phrase était « Elle l'a beaucoup remercié. », le *l'* renverrait certainement à *Paul* et non à *cadeau*. Pour résoudre ce genre d'ambiguïté, il faut compléter les algorithmes de GEOLSemantics, en ajoutant par exemple des indications, en fonction des verbes et ce qu'ils attendent comme complément (animé/inanimé, ...). Il existe des ressources dans lesquelles nous pourrions récupérer ces informations ; mais ce travail n'a pas encore été réalisé car nous nous intéressons principalement à des textes concernant des personnes et des organisations. Il s'agit donc de privilégier les référents de type personne ou organisation car c'est suffisant pour l'instant.

Par contre, aucune ambiguïté sur l'exemple *« Paul a invité Sarah à son anniversaire. Elle l'a beaucoup remercié. »*, les informations morphosyntaxiques et sémantiques indiquent que Sarah est du genre "Féminin" et de type "Personne". De ce fait, le pronom *Elle* réfère à cette personne. Le *l'* fait référence très certainement à une entité de type "Personne" à savoir *Paul*.

En ce qui concerne la gestion des modalités et de la négation, afin de ne pas encombrer la structure en arbre de l'analyse syntaxique, GEOLSemantics a choisi de convertir les relations exprimant la négation ainsi que les modalités en propriétés indiquées à l'intérieur de l'élément *head* de la relation. Par exemple, la phrase *« Je ne vais pas dormir. »* contient une seule relation Agent-Verbe avec les informations *dormir(verbe,negative,modal_aller) Je(pronoun)* [Kerdjoudj, 2015].

Une fois le processus de GAL terminé, nous obtenons un texte segmenté contenant des

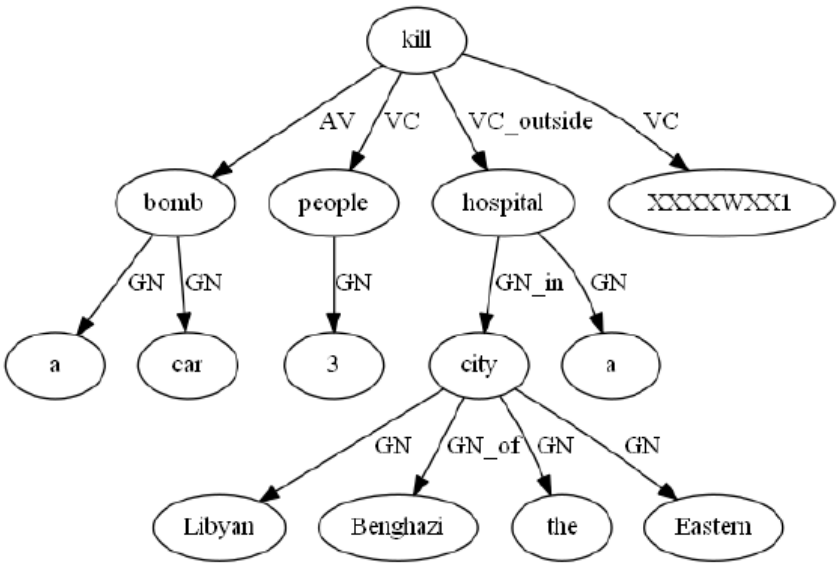



FIGURE 1.6 – Exemple de représentation d'un arbre syntaxique

lemmes catégorisés ainsi que des relations binaires indépendantes de la structure originale du texte. En sortie de ce traitement linguistique, on obtient un fichier XML. Le document est constitué de paragraphes composés de phrases qui, elles-mêmes, contiennent :

— des mots (voir la figure 1.1.1) ;

— des entités nommées (voir l'exemple 1.1.2) ;

— des entités nommées complexes (voir l'exemple 1.1.3)

— des relations (voir la l'exemple 1.1.4).

Les entités nommées peuvent être constituées d'une simple unité (un mot) ou de relations.

Les relations sont composées d'une tête, d'un dépendant, et peuvent avoir, en sus, une ou plusieurs indications linguistiques.

Cette analyse constitue une base pour créer les règles d'extraction de connaissances que nous allons voir dans la section 1.1.1 qui suit.

**Exemple 1.1.1.** Exemple de résultat généré par GAL pour les mots

```
<word>
  <posBeg>6</posBeg>
  <lemma>aller</lemma>
  <catPos index="no">+vt</catPos>
  <mCat>V</mCat>
  <prop index="no">+intransitif+indpres+3ps</prop>
  <posEnd>8</posEnd>
</word>
<word index="no">
```

```
  <posBeg>9</posBeg>
  <lemma>à</lemma>
  <catPos index="no">+prep</catPos>
  <mCat>Prep</mCat>
  <posEnd>10</posEnd>
</word>
```

**Exemple 1.1.2.** Exemple de résultat généré par GAL pour les entités nommées

```
<en entype="pers">
  <unit>
    <posBeg>0</posBeg>
    <lemma>Sarah</lemma>
    <catPos index="no">+prenom</catPos>
    <mCat>NP</mCat>
    <prop index="no">+f+pers</prop>
    <posEnd>5</posEnd>
  </unit>
</en>
<en entype="loc">
  <unit>
    <posBeg>11</posBeg>
    <lemma>Paris</lemma>
    <catPos index="no">+np</catPos>
    <mCat>NP</mCat>
    <prop index="no">+loc+town</prop>
    <posEnd>16</posEnd>
  </unit>
</en>
```

**Exemple 1.1.3.** Exemple de résultat généré par GAL pour les entités nommées complexes

```
<en entype="loc">
  <relation reltype="DG">
    <head>
      <posBeg>236</posBeg>
      <lemma>rue</lemma>
      <catPos index="no">+annp</catPos>
      <mCat>S</mCat>
      <prop index="no">+fs+loc+geo+strType</prop>
      <posEnd>239</posEnd>
    </head>
    <dept>
      <posBeg>233</posBeg>
      <lemma>23</lemma>
      <catPos index="no">+artcard</catPos>
      <mCat>J</mCat>
      <prop index="no">+numVoie+loc+geo</prop>
      <posEnd>235</posEnd>
    </dept>
  </relation>
  <relation reltype="GD">
    <head>
      <posBeg>236</posBeg>
      <lemma>rue</lemma>
      <catPos index="no">+annp</catPos>
      <mCat>S</mCat>
      <prop index="no">+fs+loc+geo+strType</prop>
      <posEnd>239</posEnd>
    </head>
```

```
    <dept>
      <posBeg>243</posBeg>
      <lemma>Gascogne</lemma>
      <catPos index="no">+np</catPos>
      <mCat>NP</mCat>
      <prop index="no">+fs</prop>
      <posEnd>251</posEnd>
    </dept>
    <lingIndication index="no">
      <posBeg>240</posBeg>
      <lemma>de</lemma>
      <catPos index="no">+prep</catPos>
      <mCat>Prep</mCat>
      <posEnd>242</posEnd>
    </lingIndication>
  </relation>
</en>
```

**Exemple 1.1.4.** Exemple de résultat généré par GAL pour les relations

```
<relation reltype="AV">
  <head>
    <posBeg>6</posBeg>
    <lemma>aller</lemma>
    <catPos index="no">+vt</catPos>
    <mCat>V</mCat>
    <prop index="no">+intransitif+indpres+3ps</prop>
    <posEnd>8</posEnd>
  </head>
  <dept>
    <posBeg>0</posBeg>
    <lemma>Sarah</lemma>
    <catPos index="no">+prenom</catPos>
    <mCat>NP</mCat>
    <prop index="no">+f+pers</prop>
    <posEnd>5</posEnd>
  </dept>
</relation>
```

### Outil d'extraction de connaissances GEX+GEXIH

Le système d'extraction de GEOLSemantics (GEX) est un outil pour extraire des informations à partir des données textuelles produites par GAL. Pour cela, il se fonde sur une ontologie, des ressources linguistiques, un générateur de règles d'extraction (dit ERG) et un module de mise en cohérence (dit GEXIH).

Le module GEX, en utilisant des règles lexicosyntaxiques, peut sélectionner les entités nommées et les relations syntaxiques afin d'extraire les informations utiles en fonction du contexte. Ceci permet de lever des ambiguïtés sémantiques restées irrésolues au niveau de GAL. Par exemple, dans le projet *SafeCity*, la détection correcte des entités nommées

de lieu est primordiale. Ceci est fait à partir de celles déjà détectées mais affinée aussi avec les entités nommées non typées à l'étape précédente ; ce qui est rendue possible par le module GEX. Ces informations sont ensuite instanciées dans l'ontologie sous forme de triplets RDF.

Le module de mise en cohérence GEXIH prend alors le relais. Il permet de résoudre les dates relatives, de regrouper et d'aligner les entités, et de regrouper les événements en considérant, non pas la phrase comme dans GEX, mais l'ensemble du texte.

L'objectif de notre extraction (cf. module GEX dans la figure 5.3) est de repérer des entités nommées et surtout des événements pouvant nécessiter l'intervention des autorités locales (pompiers, ambulanciers, policiers, agents d'entretien, etc.). Il faut caractériser ces événements le plus précisément possible, en particulier leur localisation, afin que l'information soit exploitable.

Le module GEX utilise des motifs syntactico-sémantiques guidés par une ontologie. Nous avons élaboré une ontologie permettant la modélisation des événements repérés dans les textes. Cette ontologie (cf. tableau 1.1) a été créée à l'aide de Protégé [7] au regard des spécifications du système d'alerte puis des différents corpus des événements.

| Concept | Sous-type d'action | Lieu | Date | Entité concernée |
|---|---|---|---|---|
| Incendie | event-type | event-place | event-date | isBurned |
| Accident | event-type | event-place | event-date | involve |
| Inondation | event-type | isFlooded | event-date | inundationSource |
| Colis suspect | package-type | event-place | event-date | – |

TABLEAU 1.1 – Ontologie partielle des événements

Cette ontologie fait partie d'une ontologie plus générale de la société GEOLSemantics dans laquelle nous avons créé 170 classes, 363 relations entre classes et 181 propriétés de classe. Nous avons déjà développé les définitions des entités nommées de type lieu, date, (cf. tableau 1.2) relatives aux connaissances communes à divers domaines [8].

Les patrons syntactico-sémantiques (dits *patterns* d'extraction) sont créés afin d'extraire

7. voir https ://protege.stanford.edu/
8. Cette ontologie n'est pas disponible pour des raisons commerciales

| Concept | Propriétés | Commentaire |
|---|---|---|
| Location | loc-type | type du lieu |
| | loc-name | nom du lieu |
| | locatedIn | localisé à/dans |
| | hasGeoPosition | position géographique du lieu |
| GeoPosition | coordinates | les coordonnées |
| Date | dateBeg | la date de début |
| | dateEnd | la date de fin |

TABLEAU 1.2 – Ontologie partielle pour les entités

les informations à partir des résultats de l'analyse linguistique profonde. Un pattern d'extraction est composé de trois types d' éléments :

1. un concept et une propriété définis par l'ontologie et permettant la représentation de l'information extraite au format RDF [Cyganiak *et al.*, 2014 accessed February 9 2019] ;
2. un patron lexico-syntaxique qui contient les informations morpho-syntaxiques d'une relation syntaxique ;
3. un contexte constitué d'une ou plusieurs relations syntaxiques, aidant à la désambiguïsation du sens de l'action et du type de l'entité.

Le premier élément, *i.e* le concept et la propriété, est défini par l'ontologie. Il permet de représenter l'information extraite au format RDF [Cyganiak *et al.*, 2014 accessed February 9 2019]. De ce fait, la terminologie utilisée doit être strictement identique à celle de l'ontologie définie sinon l'information ne sera pas extraite correctement par la suite dans le processus de traitement.

Le patron lexico-syntaxique, quant à lui, contient le type de relation syntaxique (soit nom-complément de nom, agent-action, etc.), suivi d'une liste de lemmes et/ou une liste de catégories et/ou une liste de propriétés sémantiques décrivant la tête de la relation syntaxique puis celle du dépendant. Notre système permet d'établir une correspondance entre le patron lexico-syntaxique et les relations syntaxiques du résultat du GAL 7.3.2. Chaque règle peut contenir un contexte constitué d'une ou plusieurs relations syntaxiques.

Enfin, le contexte est utilisé à la fois pour désambiguïser le sens de l'action et pour désambiguïser le type de l'entité. Par exemple, prenons la phrase "Nice organise une fête pour le printemps". Dans l'analyse linguistique, "Nice" est identifié comme une entité nommée de type *lieu*, mais lors de l'extraction, la relation "organiser Nice" permet de préciser que Nice désigne la mairie de Nice, qui est en fait une organisation et non un lieu.

Une fois l'étape de désambiguïsation terminée, il est possible de regrouper différentes occurrences désignant la même entité (personne, lieu, organisation, etc.)

Dans le cas de la désambiguïsation temporelle, GEXIH propose également un traitement consistant à transformer les dates relatives identifiées au niveau du module GAL comme *demain* ou *hier* en dates absolues en utilisant une date de référence. Il s'agit de la date absolue citée dans le texte ou, en cas d'absence de celle-ci, la date d'émission présente dans les métadonnées du tweet. Pour ce faire, les expressions de dates relatives sont identifiées au niveau du module GAL. Par exemple, *demain* est transformé en *XXXXXX+1* et le processus de traitement utilise la date de référence afin d'obtenir une date absolue.

## 1.2 Repositionnement de la problématique

Nous avons vu dans l'introduction que nous allions nous confronter à plusieurs verrous liés à la constitution d'un corpus sur un territoire particulier, le traitement sémantique de réseaux sociaux, l'identification d'événements et le suivi d'événements. Certains verrous nécessitent un enrichissement des ressources existantes, d'autres une certaine adaptation des algorithmes existants chez GEOLSemantics ; tandis que d'autres nécessitent la création d'outils totalement inexistants et qui font partie de domaines de recherche très actifs comme nous le verrons au chapitre 2.

En général, les projets chez GEOLSemantics se font sur des corpus fournis par le client, par exemple des textes juridiques pour le projet *Lexbase* ou du texte commercial pour le projet *Coca Cola*. Ces corpus sont fournis à chaque fois par le client au format .txt ou .pdf sur un disque dur ou sur un serveur sécurisé. Il arrive cependant que GEOLSemantics doivent récupérer le corpus sur le web comme pour le projet *demo-AFP* dans lequel il

fallait extraire et représenter par des graphes des actions dans des textes journalistiques, ceci en utilisant l'API de l'*AFP*.

Mais, dans le cadre de ce travail de recherche, comme nous allons le voir dans la section 1.4, nous devons formaliser la constitution d'un corpus spécifique qui se génère à partir de filtres exprimés par l'utilisateur final sur un réseau social, dans notre cas Twitter. Par exemple, une zone géographique d'émission de tweet pourrait être un paramètre de filtrage. Notons que ce type de corpus composé de tweets est inédit pour GEOLSemantics, tant par son volume exponentiel (des centaines de millions de tweets par jour) [9] que par la densité de l'information qui est bien moindre rapportée au volume. Ceci nous place dans un contexte de données massives (dites *Big Data*) [Delort, 2018] comme défini dans la figure 1.7.

En effet, si l'on se réfère à la figure 1.7, le carré sans nom (en bas à gauche) représente des données à faible volume et à faible densité comme un blog qui contient seulement quelques octets avec des informations utiles ou non. Mais en accumulant des données similaires, par exemple en analysant plusieurs blogs à des fins d'études sociologiques, on peut passer dans le domaine des données massives (dit *Big Data*) exploitables par accumulation dans le cadre de cet exemple.

Le carré *BD business* (en bas à droite) représente des données à forte densité en information et à faible volume comme un système d'information classique d'une entreprise. Ces bases de données représentent de quelques méga-octets à des giga-octets et contiennent jusqu'à 10% d'information utile.

Le carré *Big Data* (en haut à gauche) représente des données à fort volume et à faible densité en information. Dans le contexte de notre étude, sur les millions de tweets par jour, bien moins de 10% sont porteurs d'une information utile ; et cela peut baisser considérablement selon le domaine de recherche d'information. De plus, il faut aussi réaliser un traitement afin d'extraire l'information voulue, un traitement difficile étant donné le volume des données.

Le carré *BD courtier données* est le domaine des courtiers de données, comme les en-

9. `https://www.internetlivestats.com/twitter-statistics`

treprises Acxiom [10] ou choicePoint [11]. À titre d'exemple, Acxiom possède 1 500 points d'information sur 700 millions de personnes contenant des informations comme héritier potentiel, foyer orienté diabète, etc. ChoicePoint, quant à elle, possède 17 milliards d'enregistrements sur des personnes ou des firmes ; elle propose par exemple d'aider les casinos à détecter des relations entre joueurs et croupiers [Delort, 2018]. Il est à noter qu'aucun traitement complexe pour extraire l'information n'est à faire dans ce cas de figure puisqu'elle est souvent bien structurée et organisée.

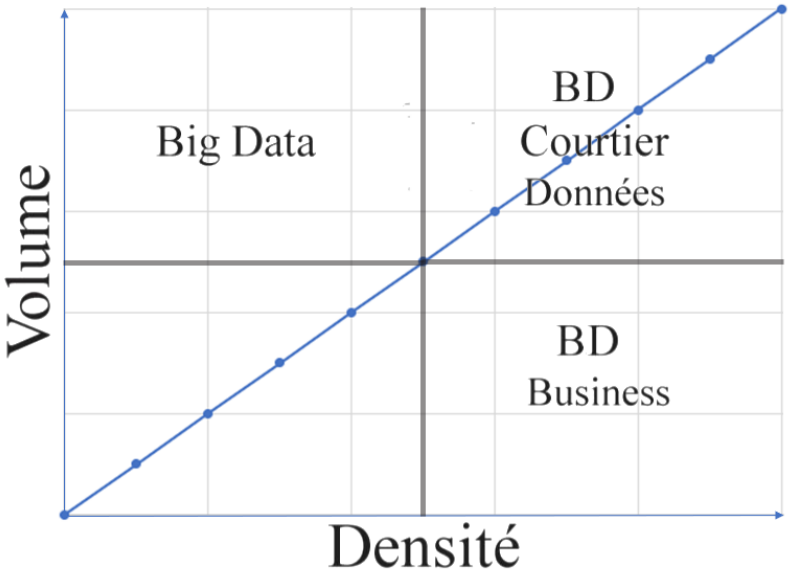



FIGURE 1.7 – L'information vue selon son volume et sa densité (source : [Delort, 2018])

Dans un second temps, il faut adapter les ressources de GEOLSemantics (dictionnaires, ontologie, règles d'extraction) pour la gestion des tweets qui sont en grande partie écrits dans un langage particulier. Une analyse linguistique spécifique doit être proposée.

De plus, comme nous nous positionnons dans la gestion d'événements pour l'émission d'alertes vers des organismes publics, il faut aussi adapter l'analyse selon les événements à extraire.

Pour ce faire, Il faut bien définir le concept d'événement dans un contexte de geo-chrono-localisation, facilitant le regroupement de tweets selon un événement. Ce regroupement va permettre d'enrichir les méta-informations de ce dernier. Il faut donc définir des critères

10. `https://www.acxiom.com`
11. `https://epic.org/privacy/choicepoint`

formels de regroupement.

Enfin, il faut développer un système de localisation qui soit assez robuste et rapide d'accès afin de localiser au plus tôt les événements détectés.

## 1.3 Champs d'expérimentation

GEOLSemantics est spécialiste dans le domaine de la linguistique et de la sémantique, avec des outils capables d'analyser des contenus textuels pour identifier, normaliser et structurer l'information (cf. figure 1.1). Les contenus structurés sont ensuite utilisés par diverses applications (par exemple, des systèmes d'aide à la décision et/ou à la prédiction, etc.). La plupart des projets de GEOLSemantics sont des projets de recherche. Dans le contexte de ce travail de doctorat concernant la géo-chrono-localisation d'événements sur Twitter, plusieurs projets vont servir de terrain d'expérimentation à savoir :

1. *SafeCity* : ce projet de géolocalisation a été mis en place afin de prévenir les autorités de l'endroit où un risque/événement se produit. C'est un système de sécurité qui va analyser les événements dans les données, localiser la zone des événements (qui peut être différente de la zone d'envoi) et les transmettre aux autorités.
2. *DRIRS* (Détection de Réseaux d'Influence dans les Réseaux Sociaux) : avec ce projet, nous analysons en utilisant des méthodes linguistiques ce qui est dit en arabe et/ou en français afin de savoir si un discours/texte est un discours d'influenceurs ou non. Il s'agit aussi de déterminer qui sont les influenceurs et les influencés dans le cadre de dialogues. Ces derniers sont des personnes qui peuvent paraître fragiles au regard des premiers. Elle risquent d'être visées plus particulièrement par les influenceurs.

L'analyse est réalisée sur des textes écrits ou des transcriptions effectuées par Vocapia [12] qui est une société de R&D spécialisée dans le développement de logiciels pour le traitement de la parole.

*Safecity* et *DRIRS* sont deux projets qui apportent une nouvelle problématique qui est celle

12. `https://www.vocapia.com`

de la gestion des réseaux sociaux. La nature des textes des réseaux sociaux est particulière. Les textes sont courts, souvent rédigés avec beaucoup de fautes orthographiques. Cet état de fait est une première pour GEOLSemantics qui a l'habitude de projets ne gérant que des textes bien écrits comme le moteur de recherche Lexbase [13] reposant sur un contenu juridique bien structuré et bien écrit.

Le travail de cette thèse sera dans sa majorité appliqué sur le projet *SafeCity*.

Certains verrous résolus vont permettre d'avancer sur le projet *DRIRS*, principalement ce qui concerne la création et le traitement des corpus de tweets.

## 1.4 Mise en perspective du projet *SafeCity*

Le développement des villes ne cesse de s'accélérer avec une urbanisation galopante dans le monde. En plus d'une population grandissante, les villes doivent faire face à des menaces de plus en plus importantes, à la fois liées aux risques naturels – dont le dérèglement climatique est une composante de plus en plus forte – et aux risques d'origine humaine, dont la criminalité et le terrorisme qui se développent dans toutes les régions du monde. Conjointement, le développement des technologies permet aux villes d'intégrer de nouveaux moyens leur permettant d'accéder au rang de villes intelligentes.

Le concept de ville intelligente nous laisse penser à un système capable de raisonner tout seul et donc dépendant de moins en moins de l'humain, servant à gérer une ville d'une manière plus ou moins automatique, en utilisant plusieurs ressources notamment les données laissées par les habitants sur les réseaux sociaux. Par exemple, l'application de l'intelligence artificielle dans une ville peut aller d'un système de caméra de surveillance qui détecte des événements à valider par des humains jusqu'à une gestion totalement autonome comme le *Ghost port* (ou port fantôme) chinois, premier port entièrement automatisé à Qingdao [14] géré entièrement avec des machines.

La prédiction d'événement ou même la détection d'un événement suffisamment tôt pour

13. `https://www.lexbase.fr/`
14. `https://news.cgtn.com/news/3d637a4e31677a4d/share_p.html`

une intervention rapide est une perspective intéressante pour ces villes intelligentes. Cette détection peut être tout aussi bien dans un système de vidéos surveillance que sur les réseaux sociaux. Certains événements nécessitant l'intervention des services de la ville peuvent être gérés plus rapidement en utilisant les réseaux sociaux. Par exemple, beaucoup plus de personnes vont prendre en photo un dépôt d'ordures sauvages et le publier sur les réseaux sociaux par rapport au nombre de personnes qui vont alerter les services concernés. Avec un système suffisamment solide, l'alerte peut être lancée automatiquement sans que personne n'appelle les services de nettoyage.

*SafeCity* est un projet de type programme d'investissements d'avenir (PIA) financé par la BPI [15] qui a pour thème « la sécurité de la ville numérique ». Commencé en 2018, la mission de GEOLSemantics dans ce projet est de détecter dans les réseaux sociaux (en particulier Twitter) des événements qui pourraient nécessiter l'intervention des autorités locales (pompiers, ambulanciers, policiers, etc.). Pour ce faire, il est nécessaire de caractériser ces événements en particulier avec leur localisation précise. Deux sites d'expérimentation sont prévus : l'un à Nice, l'autre au quartier d'affaires de la Défense.

Les événements spécifiés dans le projet sont : accident, arbre tombé, colis suspect, incendie, inondation, bagarre, dépôt d'ordures sauvage, organisation de fêtes (*rave party* sans autorisation), route coupée, etc. La nécessité d'établir un corpus pour chaque événement avec des contraintes de lieux (dans notre cas, *Nice* et *La Défense* pour l'expérimentation) constitue déjà une première contribution par le fait qu'il faut traiter un maximum de tweets émis sur le territoire concernant les événements spécifiés pour déterminer les plus pertinents.

La solution GEOLSemantics doit être modulable et facile à intégrer avec d'autres systèmes. En effet, plusieurs partenaires interagissant dans le projet *SafeCity*, différentes technologies proposées comme la gestion de foule, le traitement d'images, l'affichage 3D, etc., sont à intégrer dans le processus global de traitement linguistique.

15. `https://www.bpifrance.fr`

### 1.4.1 Tâches

Dans le cadre du projet *SafeCity*, nous devons effectuer un certain nombre de traitements afin de répondre au mieux à la problématique exposée lors de notre chapitre introductif.

#### Définition naïve du concept de géo-chrono-localisation

La localisation d'un objet est la détermination de sa position géographique. Cette localisation peut être générale (région, pays, ville, etc.) comme elle peut être réelle.

La géolocalisation est la localisation géographique réelle d'un objet. La géolocalisation implique généralement la génération d'un ensemble de coordonnées géographiques et est étroitement liée à l'utilisation de systèmes de positionnement.

Selon les normes ISO/IEC 19762 :2016 qui introduisent un vocabulaire harmonisé des techniques automatiques d'identification et de saisie de données (AIDC) [16], la géolocalisation est l'ensemble des coordonnées de latitude et de longitude d'un lieu particulier.

La chrono-localisation est la localisation spatiale et temporelle d'un objet ou d'une personne, ou, selon Joël Ferry, la localisation en temps réel [17]. La chrono-localisation est un terme qui s'est développé surtout dans le contexte moderne et le développement des téléphones mobiles [Patrix, 2013].

Dans le cadre de cette thèse, nous allons transformer du texte brut, en l'occurrence des tweets, en coordonnées GPS "géo", à un instant $t$ qui peut être dans le texte du tweet ou dans ses métadonnées (chrono).

À l'instar du *big-bang* où le temps et l'espace étaient une singularité [Hawking *et al.*, 2020], notre événement sera composé d'un triplet $< action, lieu, date >$, soit un événement géo-chrono-localisé à un instant $t$ précis et avec des coordonnées GPS réelles. Une définition plus formelle sera présentée dans la section 5.4.

16. `https://www.iso.org/obp/ui/#iso:std:iso-iec:19762:ed-1:v1:fr`
17. la-geolocalisation-dans-tous-ses-etats-3968.html

**Analyse de réseaux sociaux**

La première tâche concerne les réseaux sociaux. Elle nécessite de créer des outils et d'en adapter d'autres déjà existants. Pour ce faire, nous serons confrontés à de nombreuses problématiques dans différents champs, notamment celles sous-jacentes à la constitution de corpus pour un territoire donné et celles entourant l'analyse des tweets et leur nature textuelle particulière. Cette étude sera menée par l'approche suivante :

— développement d'un outil de récupération de tweets émis, depuis les territoires de *Nice* ou de *La Défense* prioritairement, mais assez générique pour être étendu à d'autres lieux, et pouvant concerner l'un des événements à identifier. Puis, une extraction de ces événements géo et chrono localisés sera faite ;

— prise en compte des tweets mal écrits et optimisation des traitements ;

— évaluation du degré de confiance de l'information extraite et extension à Facebook pour faire de la prévision d'événement.

**Système d'alerte et appel d'urgence**

Le système final d'émission d'alerte consistera en une analyse en temps réel des tweets émis avec une restriction géographique correspondant prioritairement aux deux lieux de test à savoir la métropole de *Nice, Côte d'Azur* et le quartier de *La Défense* ; il est noté que le système est assez générique pour s'adapter à toute zone géographique à considérer. Le système doit ensuite afficher sur une carte un événement précis et alerter les autorités.

## 1.5 Premières conclusions

À travers ce premier chapitre, nous avons commencé par faire l'état des lieux des modules intervenant dans le processus de traitement linguistique de GEOLSemantics existant, en amont du travail de recherche en relation avec cette thèse. Nous avons aussi présenté les projets qui ont constitué des défis à relever et solutionner par les modules actuels de

GEOLSemantics, et qui ont mis en évidence des problématiques de recherche jusque-là ignorées par GEOLSemantics.

Dans les chapitres suivants, nous allons aborder l'état de l'art de chaque problématique, puis la solution développée soit par l'amélioration des processus déjà présents, soit par la création de nouveaux outils. Nous allons aussi comparer au travers de différentes expérimentations, les solutions adoptées à d'autres proposées en recherche dans le même contexte. Rappelons que les traitements de GEOLSemantics sont fondés sur des ressources et des règles. Il est facile de retrouver les règles appliquées pour expliciter chaque résultat d'une analyse, contrairement aux traitement fondés sur des statistiques qui sont en grande partie une sorte de boîte noire, efficace mais intraçable dans la plupart des cas. Un système hybride est envisageable combinant l'efficacité des outils statistiques et la possibilité de justification des résultats des outils de GEOLSemantics.

# Chapitre 2

# État de l'art

## Sommaire


**2.1 Analyse des réseaux sociaux . . . . . 39**

**2.2 Extraction d'événements et localisation . . . . . 43**

**2.3 Fouille de données . . . . . 46**

**2.4 Fouille de textes . . . . . 53**


Nous allons maintenant faire un tour d'horizon des domaines de recherche qui auront un rôle central dans la résolution de notre problématique liée à la détection géo-chrono-localisée d'événements dans des tweets en temps réel ou quasi-réel. Pour ce faire, il nous a semblé opportun de connaître l'existant concernant l'analyse des réseaux sociaux, l'extraction d'événements et leur localisation, les approches inductives de l'Intelligence Artificielle (IA) par la fouille de données et plus précisément la fouille textuelle puisque nous manipulons des tweets.

## 2.1 Analyse des réseaux sociaux

Le phénomène des réseaux sociaux est celui qui a le plus radicalement remodelé la façon dont les gens obtiennent des informations et passent leur temps, depuis la création de la télévision. Il a un impact gigantesque sur le monde moderne. Facebook, qui est de loin la

plus grande plate-forme de réseaux sociaux, compte 2,3 milliards d'utilisateurs actifs dans le monde par mois. En 2016, l'utilisateur moyen passait 50 minutes par jour sur Facebook et ses plate-formes sœurs Instagram et Messenger [1].

Allcott *et al.* [2020] ont trouvé que la désactivation de Facebook pendant les quatre semaines précédant l'élection américaine de mi-mandat 2018 a provoqué :

— une activité en ligne réduite, tout en augmentant les activités hors ligne telles que regarder la télévision seul et échanger avec la famille et les amis ;

— une réduction à la fois de la connaissance des informations factuelles et de la polarisation politique ;

— un bien-être subjectif accru ;

— une forte réduction persistante de l'utilisation de Facebook après l'expérience.

A l'heure actuelle, c'est une évidence que les réseaux sociaux font partie intégrante de notre vie moderne. De plus en plus d'informations apparaissent dans ces réseaux sociaux avant même d'être produites sur les réseaux d'information classiques.

Ce phénomène intéresse de plus en plus de domaines socio-économiques, notamment par la capacité d'interaction directe avec les utilisateurs, mais aussi par le fait que l'on peut voir une opinion en temps réel. Cela a été une révolution pour le marketing d'entreprise. Appel *et al.* [2020] ont identifié les futures orientations de recherche pertinentes pour les universitaires et les praticiens dans ce domaine particulier.

Sivarajah *et al.* [2020] ont démontré, quant à eux, la capacité des approches d'analyse liées aux données massives (*big data*) et aux réseaux sociaux dans un environnement Web participatif pour permettre aux organisations B2B de devenir rentables durablement, grâce à des opérations stratégiques et des activités commerciales liées au marketing.

Ceci dit, de plus en plus de recherches s'intéressent aux réseaux sociaux dans bien d'autres domaines, avec des répercussions qui ont une portée mondiale. Par exemple, Duraisamy *et al.* [2020] et Depoux *et al.* [2020] ont réalisé une étude phénoménologique sur les réseaux sociaux, en analysant le degré d'information et de désinformation autour de la COVID-19.

1. voir Facebook-Reports-Fourth-Quarter-and-Full-Year-2018-Results

Bien que plusieurs recherches existent sur les défis et les difficultés liés à des méthodes d'analyse des réseaux sociaux, il existe peu de recherches sur les étapes concernant la découverte, la collecte et la préparation des données. Pour combler cette lacune, Stieglitz *et al.* [2018] ont mené une analyse documentaire approfondie et structurée à travers laquelle ils ont identifié les défis relevés et les solutions proposées.

Twitter est un autre réseau social qui fait l'objet de nombreuses études, puisque c'est le seul réseau social qui propose une API pour récupérer des publications faites par des utilisateurs (dans ce cas, des tweets).

Une fois les tweets récupérés, il faut les analyser et extraire les informations inhérentes automatiquement, et sans passer par les procédures habituelles comme :

— chercher avec un hashtag particulier : l'idée d'un hashtag est de regrouper tous les tweets sur un domaine particulier, mais cette procédure est peu pertinente, vu que de plus en plus de personnes utilisent un hashtag populaire pour faire une publicité sur des produits sans aucune relation avec le thème du hashtag (ce qui va polluer l'analyse) ;

— chercher avec des mots-clés : il est aussi possible de chercher des tweets en utilisant des mots-clés mais les nuances d'une langue rendent les résultats moins pertinents. Par exemple : le mot explosion peut être compris au sens littéral du terme mais aussi être une métaphore comme une explosion de joie.

Nous allons donc concevoir une architecture d'analyse capable de faire une recherche non pas simplement sur des hashtags et des mots-clés, mais sur la sémantique d'un thème donné. De ce fait, un journaliste, par exemple, pourra obtenir le premier tweet sur un événement donné, bien avant que le hashtag ne devienne populaire, et sans s'abonner à des milliers de comptes d'utisateurs de Twitter.

Mais avant d'en arriver là, il faut combiner l'analyse des réseaux sociaux avec le traitement automatique du langage afin de générer un système d'analyse fondé sur la sémantique.

Une des plus grandes problématiques du traitement automatique du langage (TAL) appliqué aux réseaux sociaux reste les fautes qu'elles soient orthographiques (cf. figure 2.1), syntaxiques (cf. figure 2.2) ou même grammaticales (cf. figure 2.3). L'orthographe est gé-

néralement très limité pour ne pas dire dévoyé. Le traitement des échanges sur les réseaux sociaux nécessite la création de règles spécifiques de traitement du langage, qui doivent être définies pour passer outre la plupart de ces fautes afin d'obtenir effectivement le sens porté par le message.




FIGURE 2.1 – Exemple d'une faute orthographique




FIGURE 2.2 – Exemple d'une faute syntaxique




FIGURE 2.3 – Exemple d'une faute grammaticale

La détection d'événements dans les réseaux sociaux intéresse de nombreux domaines comme l'analyse des réseaux sociaux ou *Social Network Analysis* (SNA), l'analyse des médias sociaux ou *Social Media Analytics* (SMA) et le traitement des événements complexes ou *complex event processing* (CEP). On peut les distinguer ainsi :

— l'analyse des réseaux sociaux aide à comprendre les dépendances entre les entités dans les données des réseaux sociaux, mais aussi à caractériser leurs comportements et leurs effets sur le réseau dans son ensemble et à travers le temps [Tabassum *et al.*, 2018] ;

— l'analyse des médias sociaux concerne le développement et l'évaluation d'outils pour collecter, surveiller, analyser, résumer et visualiser les données des médias sociaux

[Zeng *et al.*, 2010] ;

— le traitement d'événements complexes consiste à extraire des informations des systèmes de distribution de messages. Il permet aux utilisateurs d'un système de spécifier les informations qui les intéressent [Luckham and Frasca, 1998].

Dans ce travail de recherche, notre centre d'intérêt est la détection d'événements afin de déclencher des alertes auprès d'autorités comme les pompiers ou les services publics. Nous allons tirer partie des études précédentes en proposant une approche hybride couplant le TAL, l'analyse des réseaux sociaux et le traitement d'événements complexes.

## 2.2 Extraction d'événements et localisation

Selon McMinn *et al.* [2013], un événement est une action signifiante qui se produit à un moment et à un endroit précis. Une action est signifiante si elle peut être discutée dans les médias. La détection d'événements dans un média comme Twitter a montré qu'elle peut aider dans la prévention des inondations [de Bruijn *et al.*, 2017], la détection des événements locaux [Wei *et al.*, 2019], la prédiction des mouvements boursiers [Pagolu *et al.*, 2016], etc.

L'extraction d'événements à partir de textes de type journalistique fait toujours l'objet de travaux de recherche du fait du grand nombre de problèmes à régler (ambiguïté sur les entités nommées, variabilité de l'expression d'un même événement, etc.).

L'avènement des réseaux sociaux et surtout Twitter a permis d'avoir une plate-forme qui recense différents événements. En effet, malgré l'aspect limité du nombre de caractères d'un tweet, celui-ci permet de relier une information pertinente en utilisant plusieurs éléments, notamment le *hashtag* qui est un mot-clé permettant de regrouper plusieurs tweets.

La recherche dans le domaine de l'extraction d'événements a trouvé dans les réseaux sociaux une nouvelle voie d'application. Contrairement aux textes de type journalistique, les tweets ont un nombre de caractères limités, et une écriture plus phonétique qu'orthographique ; ce qui constitue déjà un défi considérable.

L'essentiel des travaux sur les réseaux sociaux sont très récents et étaient jusqu'à présent consacrés à l'analyse de sentiments sur des produits ou des personnes.

Du fait que les utilisateurs de Twitter émettent des messages (*"twittent"*) constamment à propos d'événements dont ils sont témoins, un travail sur la détection en temps réel des tremblements de terre a été mené en 2010 [Sakaki *et al.*, 2010]. L'outil a permis de lancer une alerte plus rapidement que l'agence météorologique du Japon.

Une publication de 2013 de la revue TAL[2] fait le point sur les difficultés de traitement linguistique des messages de type tweets montrant que l'état de l'art des traitements linguistiques qui portent essentiellement sur des textes écrits correctement doit évoluer pour prendre en compte les messages des réseaux sociaux [Farzindar and Roche, 2013].

Les États-Unis ont lancé il y a trois ans, avec des moyens énormes, un programme, appelé DEFT, qui vise à une extraction d'information généralisée. Le programme doit durer cinq ans et bien que les objectifs visés aient été publics (appel d'offre public), les résultats sont confidentiels étant orientés défense.

Pour ce qui concerne ce qui nous occupe dans le cadre de cette thèse à savoir l'extraction d'événements à partir de réseaux sociaux, il existe des travaux très récents [Li *et al.*, 2014]. Les événements auxquels s'intéressent ces travaux sont des événements de la vie des internautes tels que mariages, diplômes, admissions dans une école, anniversaires, etc., avec, en général, un seul attribut à associer, la personne concernée étant toujours l'émetteur du message, ce qui n'est pas toujours le cas, comme dans l'exemple : "Mon fils a eu son bac!!".

**Exemple :** "Je me suis marié à Tom" émis par @Mary21 Marriage, @Mary21, Tom

Ce travail se préoccupe des dialogues provoqués par les tweets mais de manière limitée. Il considère qu'une réponse de félicitation représente un indice de l'intérêt de l'événement exprimé dans le message d'origine.

Un autre travail [Benson *et al.*, 2011] insiste sur le fait que la limitation en longueur des messages empêche d'obtenir la totalité de l'information sur un seul message et qu'il faut

2. Traitement Automatique des Langues

en compiler plusieurs pour avoir une information complète. D'autre part, il ajoute que certaines informations ne sont pas écrites et doivent provenir d'informations extérieures, par exemple, un guide des événements locaux publié par une municipalité.

On peut aussi citer des travaux européens comme ceux de l'Open University de Milton Keynes [Choudhury and Alani, 2014] qui ajoute dans la description des événements de la vie personnelle, la notion de temps.

Le besoin d'alerter les autorités concernées lors d'un événement qui menace la sécurité des personnes intéresse beaucoup la communauté scientifique. Un travail en 2016 sur l'analyse des tweets géolocalisés pour explorer les réponses sociales face aux phénomènes météorologiques extrêmes [Cavalière *et al.*, 2016], a établi que, si une densité anormale de tweets peut être considérée comme un marqueur d'événement, ces tweets ne constituent pas un indicateur fiable pour qualifier la dynamique physique et sociale de l'événement, en raison de trois facteurs : (i) la forte activité d'individus isolés, (ii) la télé-présence aux événements ainsi que (iii) l'incomplétude de l'information.

Des travaux récents de 2018 sur la détection d'incidents de circulation en temps réel [Paule *et al.*, 2018] ont souligné le nombre insuffisant de tweets correctement géolocalisés. En effet, de toutes les métadonnées qui accompagnent un tweet concernant la géolocalisation, celle qui apparaît le plus est celle déclarée par l'utilisateur (cf. figure 2.4). Cette métadonnée n'est pas mise à jour automatiquement selon la position de l'utilisateur, et bien souvent elle est inutile puisque l'utilisateur peut ne pas la déclarer (cf. figure 2.5) ou déclarer qu'il est 'Partout' à 'Narnia' ou '#blackLivesMatters' (cf. figure 2.6), contrairement aux métadonnées renvoyées par Twitter et qui reposent sur la position GPS de l'utilisateur, qui sont bien structurées et plus au moins précises selon le consentement de l'utilisateur.

Les villes intelligentes « *Smart cities* » reposent considérablement sur des systèmes capables d'extraire des informations en temps réel pour pouvoir lancer une alerte automatiquement en se basant sur les habitudes des utilisateurs de Twitter [Al Nuaimi *et al.*, 2018].

Les données géographiques sont intéressantes aussi dans le domaine de la prédiction comme dans les travaux de Liang *et al.* [Zhao *et al.*, 2015] en 2015, qui présentent un framework




FIGURE 2.4 – Exemple de déclaration d'un lieu correcte

d'apprentissage multi-tâches pour la prédiction des événements et leur signalement sur une zone géographique. La visualisation qui est la finalité de certains travaux de recherche est aussi un domaine de recherche important comme les travaux de Wang *et al.* [Wang *et al.*, 2017]. En effet, avec une moyenne de plus de 6000 tweets par seconde en 2020, un chiffre qui augmente chaque jour[3], le tri, la restructuration et la visualisation en temps réel de cette masse de tweets est un vrai challenge [Brosset, 2017].

Pour toutes ces raisons, les recherches sur les informations spatio-temporelles exprimées dans les tweets sont de plus en plus fréquentes. Il est difficile de s'y retrouver mais grâce aux travaux de *Crowdanalyser-spatiotemporal analysis of user-generated content* [Steiger *et al.*, 2014] financés par l'état allemand du Bade-Wurtemberg, nous avons un état de l'art de tout ce qui a été fait dans le domaine jusqu'en 2014.

## 2.3 Fouille de données

L'intelligence artificielle (IA) est la capacité d'imiter les fonctions cognitives des humains par des machines [Braunschweig, 2016]. Elle a de nombreux domaines d'application notamment la robotique, le biomédical, le traitement d'image, etc.

3. voir `https://www.internetlivestats.com/twitter-statistics`




FIGURE 2.5 – Exemple de non déclaration de lieu

Dans le cas de la fouille de données (ou *data mining*), cette IA se fonde sur des approches inductives (partant des données pour trouver un modèle, par exemple). La fouille de données [Hearst, 1997] est l'extraction des connaissances à partir de bases de données (multimédia ou non), de flux de données provenant de capteurs, etc.

Une classification des techniques peut être appréhendée selon la figure 2.7.

Détaillons maintenant cette classification.

### 2.3.1 Apprentissage machine (ou *Machine learning*)

La donnée est l'or du siècle actuel ! La ruée a déjà commencé afin d'essayer de trouver la meilleure façon de la récupérer, la stocker, la traiter et en extraire les informations pertinentes. Le marché des données est tellement fructueux qu'il existe même un marché noir de la donnée dans le web profond [4]. Selon un rapport d'IBM Marketing Cloud, "10 Key marketing Trends for 2017", 90% des données disponibles dans le monde à cette date ont été créées au cours des deux dernières années seulement, avec 2,5 quintillions d'octets [5]

4. `https://www.marketingweek.com/the-online-data-black-market`
5. 1 octet x $10^{30}$




FIGURE 2.6 – Exemple de déclaration d'un lieu imaginaire

de données par jour [6].

Avec des quantités de plus en plus importantes de données disponibles, les anciens algorithmes de gestions et de traitement des données deviennent inutilisables. Il y a là de bonnes raisons de croire que l'analyse intelligente des données deviendra encore plus omniprésente en tant qu'ingrédient nécessaire au progrès technologique.

Afin d'avoir une analyse intelligente dans un tel environnement volatile, un système doit avoir la capacité de s'adapter et d'apprendre. Si un système peut apprendre, il peut aussi faire de la prédiction, voire de la recommandation, et trouver des solutions sur différentes situations possibles [Alpaydin, 2020].

C'est ici le règne de l'utilisation de l'apprentissage automatique ( ou *machine learning*). Selon Alpaydin [2020], le *machine learning* est un traitement de données visant à construire un modèle simple qui pourrait avoir des utilisations précieuses en cas d'évolution des données ou sur d'autres données similaires. C'est un champ d'études qui se fonde sur des approches mathématiques et statistiques. Il est composé de plusieurs modèles comme les réseaux de neurones artificiels [Abiodun *et al.*, 2018], les arbres de décision [Safavian and

6. https://www.mediapost.com/publications/article/291358

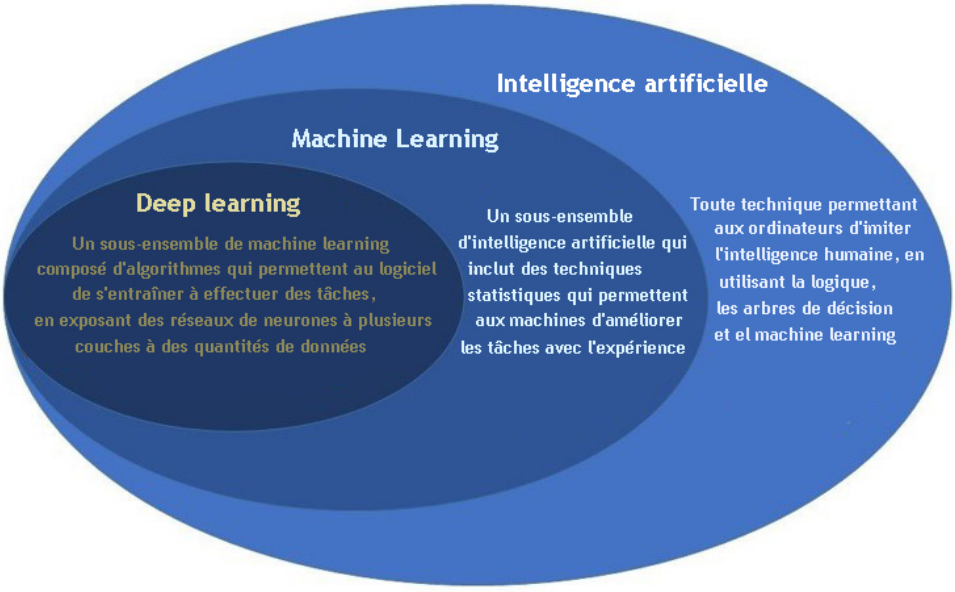



FIGURE 2.7 – Sous-ensemble de l'I.A.

Landgrebe, 1991], les machines à vecteurs de support [Nayak *et al.*, 2015], l'analyse des régressions [Chatterjee and Hadi, 2015], les réseaux bayésiens [Scanagatta *et al.*, 2019], les algorithmes génétiques [Cantú-Paz, 1998].

On trouve dans l'apprentissage automatisé, l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage semi-supervisé qui est un mix entre les deux premiers types [Zhu and Goldberg, 2009].

**L'apprentissage supervisé** : c'est un apprentissage dans lequel l'algorithme est guidé en lui fournissant des exemples probants des résultats attendus et préalablement étiquetés. L'algorithme apprend alors de chaque exemple en ajustant ses paramètres afin de diminuer l'écart entre les résultats attendus et les résultats obtenus [yves Gerlat, 2019]. Cet apprentissage a besoin d'une grande quantité de données étiqueté pour être efficace. Des exemples d'utilisation possible seraient la reconnaissance d'une photo de chien après avoir appris sur des millions de photos de chiens ou la traduction d'un texte d'une langue à une autre après avoir appris sur des millions de phrases traduites.

**L'apprentissage non supervisé** : c'est un apprentissage dans lequel la machine apprend de façon autonome. Ce qui est communiqué à la machine sont les données et la

sortie attendue. L'apprentissage non supervisé est souvent associé la notion de classification (ou *clustering*) qui est un procédé destiné à regrouper un ensemble d'éléments hétérogènes sous forme de sous-groupes homogènes ou liés par des caractéristiques communes [Laloë, 2009]. Cette technique peut être utilisée dans l'identification de fausses nouvelles, le filtrage antispam ou l'identification d'activités frauduleuses ou criminelles.

Le problème de classification a été abordé dans de nombreux contextes et par des chercheurs de nombreuses disciplines. Cela reflète son attrait général et son utilité comme l'une des étapes de l'analyse des données. Jain *et al.* [1999] ont présenté un aperçu des méthodes de classification dans une perspective de reconnaissance statistique de formes afin de fournir des conseils utiles et des références aux concepts fondamentaux accessibles à la communauté des chercheurs.

Au cours des deux dernières décennies, l'apprentissage automatique est devenu l'un des piliers de la technologie de l'information. L'apprentissage supervisé est sans doute le type d'apprentissage automatique le plus répandu et qui a beaucoup de succès dans divers domaines d'application [Patel *et al.*, 2015]. Cependant, de nombreuses méthodes d'apprentissage supervisé font l'hypothèse commune que les données d'apprentissage et de test sont tirées de la même distribution. Quand cette contrainte est violée, un classificateur entraîné sur le domaine source subira probablement une baisse en performances lors des tests sur le domaine cible en raison des différences entre les domaines [Wilson and Cook, 2020a]. Cette problématique est résolue grâce à l'apprentissage par transfert, ou *transfert learning*, (détaillé dans la section 2.3.3), qui est une composante de l'apprentissage profond, ou *deep learning*, (voir section 2.3.2).

### 2.3.2 Apprentissage profond (ou *Deep learning*)

Un réseau de neurones artificiels est un système inspiré du fonctionnement des neurones biologiques sur lequel est fondé l'apprentissage profond (*Deep learning*) qui est quant à lui un modèle utilisant plusieurs couches de neurones pour extraire progressivement des entités de niveau supérieur de l'entrée brute. Il est notamment très utilisé dans la

reconnaissance d'image. Pour entraîner le réseau, on lui fournit un grand nombre d'images pour lui apprendre à reconnaître quelque chose dans une image, par exemple, un pont. Les images fournies soit contiennent l'objet recherché, soit tout autre chose. Ces images sont converties en données sur lesquelles les neurones vont affecter des poids indiquant la présence ou non de l'objet. Toujours est-il que c'est la couche neuronale finale qui par accumulation d'information va déduire ou non la présence de l'objet recherché. Une fois cette étape faite, le réseau de neurone va comparer avec les résultats de l'annotation humaine sur le même ensemble d'image. C'est cette mémorisation des bons et mauvais résultat qui va lui permettre de déduire la présence ou non d'un objet dans les nouvelles images à analyser avec un pourcentage d'erreur, du coup, beaucoup plus faible.

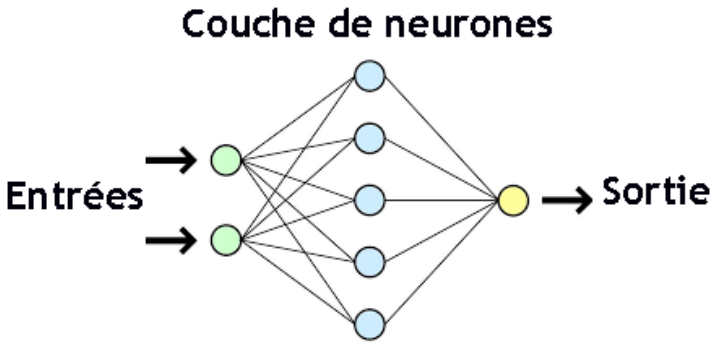



FIGURE 2.8 – Schéma d'un réseau de neurones à une couche

Dans certains cas, le *Deep learning* n'est pas applicable car nous ne disposons pas de corpus assez grand pour entraîner convenablement le réseau de neurones. Dans ce cas, l'apprentissage non supervisé [Zhang and Yang, 2017; Wilson and Cook, 2020b] (ou *unsupervised learning*) est plus approprié. C'est le cas dans le contexte de notre étude car il n'existe pas de grand corpus de tweets et nous devons analyser les tweets à la volée pour déclencher une alerte au plus tôt. Il repose sur peu de données non étiquetées. Les réseaux de neurones travaillent alors sur la base de motifs, dit *pattern*, de détection.

### 2.3.3 Apprentissage par transfert (ou *Transfer learning*)

Une tendance indéniable dans l'apprentissage automatique est l'utilisation accrue des réseaux de neurones profonds lorsqu'ils sont formés sur de grandes quantités de données, ces

réseaux de neurones à plusieurs couches peuvent apprendre représentations hiérarchiques puissantes [Long *et al.*, 2015].

Le *Deep learning* permet de réaliser de l'apprentissage par *transfer learning* qui consiste à stocker les connaissances acquises tout en résolvant un problème et en l'appliquant à un problème différent mais connexe.

L'adaptation de domaine, ou *transfer learning*, fait référence à un objectif d'apprentissage où, à partir de données étiquetées dans un domaine source, le modèle d'un concept fonctionne bien aussi sur un domaine cible ; ce modèle étant différent mais relié dans le domaine cible [Goodfellow *et al.*, 2016].

En raison de sa capacité à adapter les données étiquetées pour une utilisation dans une nouvelle application, l'adaptation de domaine peut réduire le besoin de données étiquetées qui sont coûteuses dans le domaine cible. Par exemple, chez GEOLSemantics, l'écriture des règles d'extraction se fait toujours manuellement par les étapes suivante :

— récupérer d'un corpus sur un domaine donné ;

— décider des éléments à extraire ;

— créer des concepts dans l'ontologie (s'ils n'y sont pas déjà) ;

— créer les règles d'extraction les plus générales possible selon les exemples ;

— tester et modifier, supprimer, regrouper les règles selon les régressions sur les anciennes règles.

En appliquant le *transfer learning*, cette procédure peut être automatisée. Un modèle peut être entraîné sur les patrons des règles déjà écrites sur l'événement « incendie ». Ensuite, on pourrait lui demander de créer les règles sur un corpus récupéré sur l'événement « inondation » avec une reconnaissance de patrons [Bishop, 2006]. Par exemple, BERT qui est une infrastructure logicielle de *machine learning* pour le traitement automatique du langage, ou plus précisément la compréhension automatique du langage (dite NLU [7]) [Allen, 1995] (voir la figure 2.7) a la capacité de faire du *transfer learning* pour mettre en place nos nouvelles règles.

7. Natural Language Understanding

Ces trois formes d'apprentissage ont des prolongements dans le cas particulier de la fouille de texte.

## 2.4 Fouille de textes

La fouille de texte (ou *Text mining*) est un cas particulier de la fouille de données puisqu'elle ne traite que des bases de données textuelles ou documentaires [Feldman and Dagan, 1995]. C'est le processus d'extraction de connaissances ou de modèles intéressants et non triviaux à partir de documents texte.

Elle est aussi parfois considérée comme une extension de l'exploration de données ou de la découverte de connaissances à partir de bases de données [Fayyad *et al.*, 1996] pour extraire automatiquement des informations de différentes ressources écrites ; le but étant toujours de construire des modèles génériques de compréhension des textes afin d'en extraire les informations les plus pertinentes [Hogenboom *et al.*, 2010]. Bien souvent, ces analyses sont faites pour faire ensuite de la prédiction ou de la recommandation.

À l'origine, la différence principale entre la fouille de données et la fouille de texte était que, dans la fouille de texte, les modèles étaient extraits d'un texte en langage naturel contrairement à la fouille de données qui était sur des bases de données structurées [Hearst, 1999]. Bien évidemment, la fouille de données a beaucoup évolué puisqu'elle doit prendre en compte des données hétérogènes, structurées ou non, provenant de sources diverses, notamment avec l'avènement de l'Internet des objets. En 2004, M. Gancarski et Trousse parlent d'ailleurs de fouille de données complexes[8].).

Un exemple typique de fouille de données consiste à utiliser les modèles d'achat des consommateurs pour prédire les produits à placer les uns à côté des autres sur des étagères. Par exemple, si vous achetez une lampe de poche, vous achèterez probablement des piles en même temps [Hearst, 2003].

La fouille de données implique des méthodes à l'intersection de l'apprentissage automatique, des statistiques et des systèmes de base de données [Chakrabarti *et al.*, 2006]. Elle

8. `https://www.egc.asso.fr/wp-content/uploads/egc2004_atelier_fdc.pdf`

est utilisée dans différentes classes de tâches comme la détection d'anomalies, le résumé automatique de texte, le partitionnement de données, la classification, etc.

La fouille de données textuelles, quant à elle, repose sur trois techniques principales [Kerdjoudj, 2015] qui sont :

— les approches symboliques : ces approches sont à base de dictionnaires et de règles d'extraction ou patrons sémantiques ;

— les approches statistiques : elles sont à base d'apprentissage sur des textes annotés ;

— les approches mixtes : approches qui reposent sur les deux premières citées, comprenant une phase d'étiquetage puis une phase de reconnaissance par apprentissage statistique pour finir par une phase de ré-étiquetage pour combiner le tout.

Les anciennes techniques d'apprentissage statistique sont axées sur l'apprentissage des représentations de mots indépendamment du contexte, comme GloVe [Pennington *et al.*, 2014] ou Word2Vec [Mikolov *et al.*, 2013b].

Word2Vec utilise un modèle de réseau neuronal [9] pour apprendre les associations de mots à partir d'un grand corpus de textes. Il représente ensuite chaque mot par un vecteur qui sera utilisé pour indiquer le niveau de similitude sémantique entre les mots.

**Exemple 2.4.1.** Soient les mots *homme*, *femme*, *roi* et *reine* dans un espace à deux dimensions comme indiqué dans la figure 2.9), il apparaît que les vecteurs représentant les mots *homme* et *femme* sont proches l'un de l'autre, de même pour les vecteurs représentant les mots *roi* et *reine* [10].

Les nouvelles techniques d'apprentissage statistique sont axées sur l'apprentissage des représentations de mots dépendant du contexte. ELMo [Peters *et al.*, 2018] utilise un modèle de langage bidirectionnel, tandis que CoVe [McCann *et al.*, 2017] utilise la traduction automatique pour intégrer des informations contextuelles dans des représentations de mots.

BERT (pour *Bidirectional Encoder Representations from Transformers*) ou représentations d'encodeur bidirectionnel à partir de transformateurs [Devlin *et al.*, 2018a] est un modèle

9. Rappelons que les réseaux de neurones sont des systèmes informatiques inspirés des réseaux de neurones biologiques d'un cerveau [Chen *et al.*, 2019].

10. `https://www.smalsresearch.be/nlp-modeles-de-langue/`

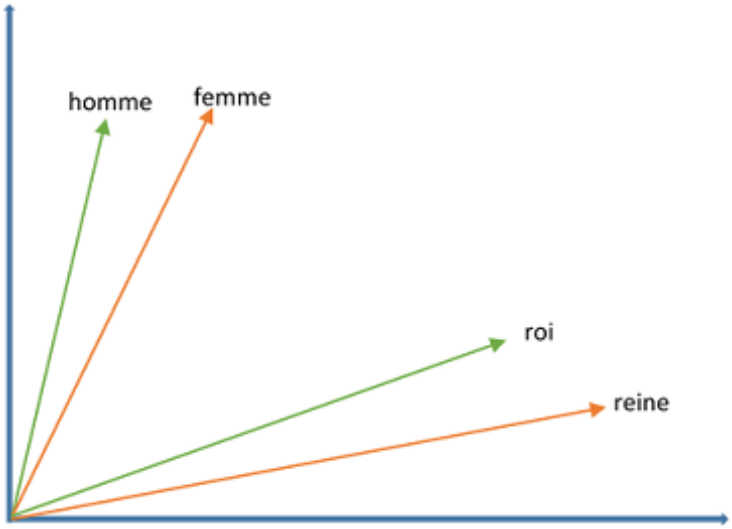



FIGURE 2.9 – Représentation *word2vec*

de représentation de mot contextualisé, fondé sur un modèle de langage masqué et pré-entraîné à l'aide de transformateurs bidirectionnels [Vaswani *et al.*, 2017]. Il existe toute une famille de modèles de représentation de langage inspirés de BERT et pré-entraînés.

Un mot n'a de sens que s'il est utilisé dans un contexte particulier. il est difficile de garder une trace des différentes parties d'une phrase, plus la phrase est longue plus c'est difficile.

Les anciens modèles de langage (tels que Word2Vec et Glove2Vec)2.10 ont construit des plongements de mots (ou *word embeddings*) sans contexte. BERT cherche le contexte.

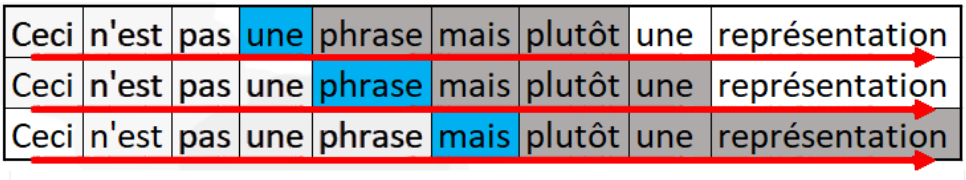



FIGURE 2.10 – Modèle de langue unidirectionnel

La plupart des modèles de langage sont unidirectionnels. Ils peuvent parcourir la fenêtre contextuelle du mot de gauche à droite uniquement ou de droite à gauche, mais pas les deux en même temps. BERT est différent : il utilise une modélisation de langage bidirectionnelle (ce qui est une PREMIÈRE). BERT peut voir la phrase ENTIÈRE de chaque côté d'un modèle de langage contextuel de mot, et presque tous les mots à la fois.2.11 formation seo [2020]

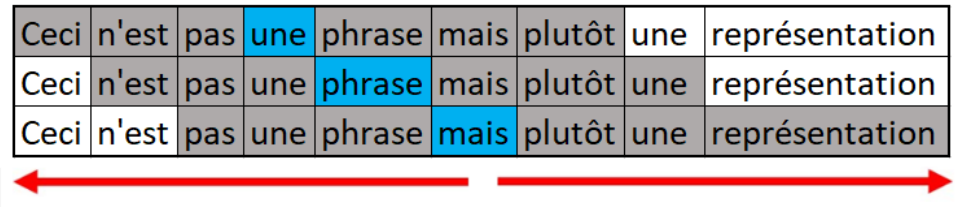



FIGURE 2.11 – Modèle de langue bidirectionnel

BERT est un modèle de langage libre entraîné sur plusieurs domaines comme la reconnaissance des entités nommées et qui peut être transposé à bien d'autres domaines ce qui est le but du *Transfer Learning*. Citons BioBERT qui est un modèle de représentation du langage pré-entraîné pour le domaine biomédical [Lee *et al.*, 2019] ou CamemBERT qui est un modèle linguistique de pointe pour le français [11].

GEOLSemantics suit une approche à base de dictionnaires et de règles d'extraction soutenues par une ontologie, mais compte remplacer certaines parties du système par une approche à base d'apprentissage pour avoir un système mixte, afin d'obtenir un système robuste, rapide, et qui ne soit pas une boîte noire comme le sont souvent les systèmes basés sur une approche statistique. Pour cela, nous allons comparer certains modules avec une approche qui repose sur le système BERT 5 car cela semble prometteur d'avoir un système hybride GEOL/BERT/ontologie pour avoir une meilleure précision et un meilleur rappel, et améliorer ainsi la détection d'événements à des fins d'alerte.

BERT est un outil de pointe dans le domaine des modèles de langage pré-entraînés. Il est très efficace dans de nombreuses tâches de compréhension de la langue comme la classification de texte. Celle-ci consiste à affecter des catégories prédéfinies à une séquence de texte donnée [Sun *et al.*, 2019]. BERT est bénéfique pour la classification de textes et d'autres tâches du TAL, ce qui peut éviter d'entraîner un nouveau modèle à partir de rien.

Sun *et al.* [2019] propose une solution générale pour perfectionner le modèle BERT. Affiner ce modèle (*fine-tune*) consiste à réaliser de petits ajustements afin d'obtenir la meilleure performance possible.

La figure 2.12 présente les trois méthodes générales proposées par Sun *et al.* [2019] à savoir :

11. `https://camembert-model.fr/`

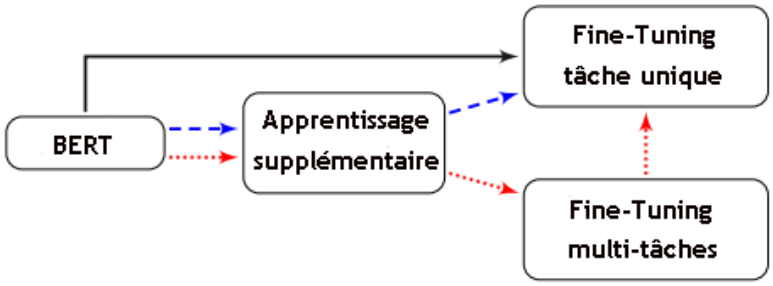



FIGURE 2.12 – Trois méthodes générales pour affiner BERT [Sun *et al.*, 2019]

— **Fine-Tuning tâche unique** (*cf.* flèche noire de la figure 2.12 ) : il existe de nombreuses façons d'utiliser BERT. Par exemple, les différentes couches de BERT capturent différents niveaux de sémantique et d'informations syntaxiques, il faut choisir la meilleure couche pour une tâche unique et la meilleure optimisation d'algorithmes et le taux d'apprentissage.

— **Apprentissage supplémentaire + Fine-Tuning tâche unique** (flèche bleue) : BERT est déjà entraîné dans le domaine général, qui peut avoir une distribution de données différente de celle du domaine cible. Il faut donc faire un apprentissage supplémentaire avec les données du domaine cible.

— **Apprentissage supplémentaire + Fine-Tuning multi-tâches + Fine-Tuning tâche unique** (flèche bleue) : lorsqu'il y a plusieurs tâches disponibles dans un domaine cible, le Fine-Tuning multi-tâches apporte encore des avantages pour affiner BERT sur toutes les tâches simultanément.

Nous avons présenté quelques techniques de fouille de textes, en détaillant notamment le système BERT et la méthode proposée par Sun *et al.* [2019] que nous avons utilisée pour notre comparaison présentée dans le chapitre 7. L'état de l'art sur BERT a été réalisé sur des points que nous avons jugés important pour notre travail. Nous suggérons aux lecteurs de lire l'article initial sur BERT [Devlin *et al.*, 2018a] [12] afin de saisir l'impact qu'il réalise sur les différents domaines.

12. meilleur article long lors de la conférence NAACL (North American Chapter of the Association for Computational Linguistics (NAACL) 2019, `https://medium.com/syncedreview/naacl-2019-google-bert-wins-best-long-paper-60e326c47a81`

Nous avons considéré BERT afin de faire une comparaison avec notre propre solution. Nous sommes conscients que le *machine learning* est ce qui est le plus proche de la perfection dans le domaine de la fouille de texte mais il reste une boîte noire qui ne permet pas d'expliquer le résultat obtenu. Or, nous voulons un système efficace mais avec une possibilité de retracer les bugs, comprendre les résultats et maîtriser chacune des étapes composant notre outil.

# Deuxième partie

# Traitement sémantique des réseaux sociaux de type tweets

# Chapitre 3

# Désambiguïsation de termes

## Sommaire


**3.1 Notions liminaires . . . . . 62**

**3.2 Méthodes de désambiguisation . . . . . 65**

**3.3 Solutions de désambiguïsation proposées . . . . . 67**

**3.4 Conclusion . . . . . 79**


Dans toute méthode de traitement linguistique, la désambiguïsation des termes utilisés dans un texte reste une phase très importante. En effet, un mot a souvent plusieurs sens qui peuvent varier du tout au tout selon son contexte d'utilisation. Par exemple, « *Il a trouvé un avocat* » reste ambigu si l'on ne connaît pas le contexte de cette phrase. Parle-t'on du professionnel ou du fruit ?

Il existe plusieurs types d'ambiguïtés, et diverses méthodes de désambiguïsations pour celles-ci, allant de l'exploitation de dictionnaires à l'utilisation d'apprentissages supervisés ou non supervisés.

Dans ce chapitre, après un rapide tour d'horizon des différents types d'ambiguités possibles et des méthodes de désambiguisation existantes, seront exposées les solutions utilisées chez GEOLSemantics pour la désambiguïsation et comment elles ont été améliorées au cours de ce travail de recherche. Un cas d'étude sur l'arabe dialectal sera exposé.

## 3.1 Notions liminaires

Dans cette section, nous commençons par décrire quelques notions utiles pour comprendre ce que l'on entend par la désambiguïsation de façon générale, puis plus particulièrement au sein de GEOLSemantics compte-tenu de ses domaines de recherche.

La désambiguïsation lexicale, ou *"Word Sense Disambiguation"* (WSD), est un domaine de recherche très important pour la communauté du traitement du langage naturel (TAL). En effet, l'attribution d'un sens particulier à une unité lexicale polysémique dans un contexte donné n'est pas chose facile.

**Définition 1** (**Désambiguïsation sémantique**). *Selon [Kelly and Stone, 1975], la désambiguïsation sémantique est le fait de déterminer l'ensemble des sens que peut prendre chaque mot dans une langue donnée.*

Chez GEOLSemantics cela consiste en l'utilisation des règles lexicosyntaxiques pour sélectionner les entités nommées et les relations syntaxiques afin d'extraire les informations utiles en fonction du contexte.

**Définition 2** (**Désambiguïsation lexicale**). *C'est la détermination du sens d'un mot dans une phrase lorsque ce mot peut avoir plusieurs sens possibles [Jacquemin, 2003]. Elle permet de décider du sens des unités lexicales dans un texte [Wilks and Stevenson, 1996].*

Elle nécessite une désambiguïsation sémantique qui sera réalisée grâce à la tâche d'association d'un mot ambigu dans un contexte donné à sa signification correcte [Singh Chaplot and Salakhutdinov, 2018].

La problématique de GEOLSemantics est d'identifier les unités lexicales d'entités nommées [Kerdjoudj, 2015]. Pour cela, nous suivons deux méthodes proposées par McDonald [1993] :

— via des preuves internes ; ce qui est propre à l'entité lexicale elle-même, grâce notamment aux majuscules.

— via des preuves externes à partir de déclencheurs tels que Ville, société, Madame, etc.

Ce type de désambiguïsation a plusieurs applications, notamment dans l'analyse des sentiments, dans la recherche d'informations (IR), dans la traduction automatique, etc.

Elle est de plus en plus importante pour aller au-delà du traitement automatique du langage vers une compréhension automatique des langues. La compréhension du langage naturel, ou *"Natural-language understanding"* (NLU), est le passage du texte lui-même vers le sens qu'il porte [Semaan, 2012]. Il s'agit d'un sous-thème du traitement du langage naturel.

Il nous semble évident, lorsqu'il s'agit de comprendre le langage naturel, qu'il est très important de rechercher le sens du mot en fonction du contexte dans lequel il est placé. Aussi, afin de gérer le sens du texte à travers une analyse linguistique, il faut une procédure visant à identifier la signification des unités du texte.

Il existe d'autres types de désambiguïsation. Dans notre travail de recherche, nous allons nous concentrer sur :

— la désambiguïsation sémantique,

— la désambiguïsation géographique,

— la désambiguïsation phonétique.

**Exemple 3.1.1** (**Ambiguïté sémantique détectée dans le cadre du projet Safe-City**)**.** Le terme *inondée* est un mot-clé pour les inondations mais dans certains contextes, il joue le rôle de métaphore comme dans la phrase *la place est inondée de supporteurs du PSG.*

**Définition 3** (**Désambiguïsation géographique**)**.** *La référence à un lieu dans l'espace géographique peut être formelle avec des coordonnées spatiales, ou informelle par l'emploi de la langue naturelle avec l'utilisation de toponymes*[1]*. Un toponyme peut représenter plusieurs lieux géographiques (par exemple, Paris au Texas et Paris en France). La désambiguïsation géographique sert à lever cette ambiguïté [Bensalem and Kholladi, 2009].*

*Selon Cuxac* et al. *[2019], la désambiguïsation géographique porte essentiellement sur des termes liés à la toponymie.*

Par exemple, la chaîne de caractères *France* peut être associée au pays ou à une personne. La compréhension du contexte permet des désambiguiser sans se poser de questions, pour un ordinateur cela est beaucoup plus complexe.

1. soit des noms de lieux

Pour GEOLSemantics, cette désambiguïsation est encore plus poussée. Dans l'exemple « *je suis près de la tour Carpe Diem pas loin du boulevard circulaire* », le lieu pointé devra être celui reconnu par un humain et donc correspond à la zone géographique entre le boulevard circulaire et la tour Carpe Diem.

Il existe plusieurs types de désambiguïsation géographique à savoir :

— désambiguïsation de lieux potentiels selon les entités nommées et/ou les mots inconnus ;

— désambiguïsation des multi-lieux : par exemple, la phrase *Je suis entre Paris et Lyon*,

— désambiguïsation selon une préposition ; par exemple, dans le groupe de mots *un incendie aux alentours de la gare Montparnasse*, l'événement n'est pas dans la gare mais dans un lieu autour de cette dernière.

**Définition 4 (Désambiguïsation phonétique).** *Selon Kévin Vythelingum [2018], la désambiguïsation phonétique est la désambiguïsation des variantes des différentes séquences de couples (graphèmes, phonèmes) possibles selon les graphèmes du mot grâce au modèle de langage.*

GEOLSemantics construit des modèles de langage sur des langues différentes spécialement l'arabe et le chinois. Ces modèle sont fournis à son partenaire Vocapia [2]. Ce dernier va utiliser ces modèles pour entraîner un système de translittération automatique de l'audio dans ces langues.

Remarquons qu'il existe aussi la phonétique auditive qui va devoir être traitée à l'avenir puisque les applications comme Twitter ont prévu d'intégrer l'audio comme moyen de communication [3]. Ceci ne sera pas pris en considération dans ce mémoire car Twitter n'a pas encore activé cette possibilité donc aucun corpus n'est disponible.

Dans ce travail, ce que nous considérerons comme désambiguïsation phonétique est la correction des fautes d'orthographe réalisées à cause du son d'un mot comme *"éléfon"* à la place de *éléphant*, ou *parler* à la place de *parlé*. La solution mise en place sera discutée dans la section 3.3.3.

2. `https://www.vocapia.com`
3. `https://blog.twitter.com/en_us/topics/product/2020/your-tweet-your-voice.html`

## 3.2 Méthodes de désambiguisation

### 3.2.1 Désambiguïsation sémantique

L'assignation d'un sens particulier à une unité lexicale exploite deux informations principales à savoir le contexte d'apparition des occurrences de chaque mot et une ou plusieurs bases de connaissances externes qui permettent de mettre en rapport les mots en contexte avec leur sens [Jacquemin, 2003].

Pour la désambiguïsation sémantique, il existe deux approches qui sont l'approche supervisée (*supervised approach*) et l'approche fondée sur les connaissances (*knowledge-based approach*).

#### Approche supervisée

La désambiguïsation supervisée du sens des mots présente le problème comme une tâche d'apprentissage automatique classique, dans lequel, premièrement, un modèle pour chaque mot est formé à partir de phrases annotées avec des sens de mot. Ces modèles sont ensuite utilisés pour désambiguïser chaque mot cible dans des phrases non utilisées pour l'apprentissage [Yuan *et al.*, 2016].

#### Approche fondée sur les connaissances

Les approches fondées sur les connaissances exploitent les informations structurelles contenues dans une ressource de connaissances lexicales sémantiques à usage général comme *WordNet*[4] pour identifier le sens correct d'un mot cible dans un contexte, c'est-à-dire une phrase ou un texte [Oram, 2001].

Les informations sur le contexte nécessitent une ou plusieurs bases de connaissances externes qui permettent de mettre en rapport les mots en contexte avec leur sens. Pour cela, il existe deux visions à savoir :

4. `https://wordnet.princeton.edu/`

— la désambiguïsation du sens des mots axée sur la connaissance (dite *knowledge-driven*) : cette méthode privilégie des ressources d'un ordre plutôt lexical pour fournir ces données ;

— la désambiguïsation du sens des mots axée sur un corpus (dite *corpus-based*) : cette méthode privilégie un corpus aux unités lexicales préalablement désambiguïsées.

Il est à noter que Pasini and Navigli [2020] a mis des bases de connaissances au service de systèmes supervisés et a réussi à tirer avantage des deux approches.

### 3.2.2 Désambiguïsation géographique

Désambiguïser les entités géographiques dans les travaux liés au traitement automatique de la langue est une tâche très importante. Par exemple, *Paris* peut être une ville en France, aux États-unis ou même une personne.

Selon Bougriou and Milat [2016], la désambiguïsation des toponymes est devenue une tâche primordiale dans plusieurs domaines tels que le web sémantique et le traitement automatique du langage naturel.

Geonames [5] permet de construire des alignements multiples en filtrant sur le continent, le pays ou le type d'entité, mais lever une ambiguïté reste une tâche difficile.

Cuxac *et al.* [2019] ont présenté une approche automatique visant à désambiguïser et aligner des entités géographiques avec une méthode basée sur des plongements lexicaux (ou *words embeding*) comme [Iacobacci *et al.*, 2016] et [Foppiano and Romary, 2018] qui permettent à partir d'un apprentissage non supervisé de lever l'ambiguïté sur un terme polysémique.

Habib and van Keulen [2011] ont mené des expériences avec un ensemble de descriptions de résidences de vacances dans le but d'extraire et de lever l'ambiguïté sur des toponymes. Ils ont montré que l'extraction et la désambiguïsation des entités nommées peuvent se renforcer mutuellement.

5. `http://www.geonames.org/`

Martineau *et al.* [2010] ont montré que l'utilisation d'un contexte syntaxique étendu et de ressources à grande échelle pouvait être très utile dans la tâche de la reconnaissance des entités nommées, tandis que Bougriou and Milat [2016] ont appliqué des méthodes de classification non supervisé pour la même tâche.

### 3.2.3 Désambiguïsation phonétique

La désambiguïsation phonétique est aussi une tâche très importante dans notre contexte d'étude, d'autant plus que le traitement à appliquer se fait sur un corpus de tweets et que bien souvent, les utilisateurs ont tendance à écrire comme ils entendent. Il a même été démontré l'existence d'une corrélation entre la phonétique et la catégorie grammaticale [Lohmann and Conwell, 2020].

La désambiguïsation phonétique peut lever plusieurs verrous concernant la désambiguïsation des noms propres mal orthographiés, la désambiguïsation des catégories grammaticales, etc.

## 3.3 Solutions de désambiguïsation proposées

### 3.3.1 Désambiguisation sémantique

La désambiguïsation sémantique chez GEOLSemantics passe tout d'abord par une désambiguïsation syntaxique. GEOLSemantics utilise une méthode d'apprentissage introduite en 1973 par le directeur scientifique de GEOLSemantics, M. Christian Fluhr [Andreewsky and Fluhr, 1973]. Cette méthode permet de constituer un algorithme capable de lever les ambiguïtés grammaticales du langage à partir d'un texte donné.

Pour cela, le texte est étiqueté entièrement en associant à chaque mot la catégorie grammaticale correspondante, grâce à l'utilisation d'un dictionnaire de formes fléchies, ainsi qu'un traitement particulier pour les mots inconnus de celui-ci. Par exemple[6], la phrase *La montre en or de mon oncle est ancienne* sera découpée en :

6. les 2 exemples suivants sont ceux de [Andreewsky and Fluhr, 1973]

— LA -> ARTICLE

— MONTRE -> SUBSTANTIF

— EN -> PREPOSITION

— OR -> SUBSTANTIF

— DE -> PREPOSITION

— MON -> ARTICLE POSSESSIF

— ONCLE -> SUBSTANTIF

— EST -> VERBE CONJUGUE

— ANCIENNE -> ADJECTIF

En cas d'ambiguïté, la phrase est annotée avec toutes les interprétations possibles. Par exemple, la phrase *Le pouvoir me serait agréable* sera déclinée en :

— LE -> ARTICLE

— LE -> PRONOM PREVERBAL

— POUVOIR -> SUBSTANTIF

— POUVOIR. -> INFINITIF

— ...

Un dictionnaire est créé qui accumule au fur et à mesure les diverses valeurs grammaticales possibles des unités.

La désambiguïsation morpho-syntaxique se passe au niveau du module GAL (cf. figure 1.5) et utilise les ressources suivantes :

— un ensemble de catégories grammaticales positionnelles, spécifiques à chaque langue, qui donnent des indications supplémentaires sur les positions des mots ; par exemple *+adja* indique les adjectifs antérieurs, c'est-à-dire situés avant un nom, alors que *+adjp* indique les adjectifs postérieurs, c'est-à-dire situés après un nom ;

— un corpus d'apprentissage, étiqueté semi-automatiquement avec les catégories positionnelles précisées dans le premier point.

Avant de commencer l'analyse, le module GAL utilise le corpus d'apprentissage pour créer les suites de bigrammes et trigrammes de catégories positionnelles présents dans les corpus, et les couples *"mot/catégorie positionnelle"* associés à leur nombre d'occurrences.

L'analyse, ensuite, commence par appliquer les bigrammes et trigrammes des catégories positionnelles ainsi que les couples (mot/catégorie positionnelle) afin de trouver la solution la plus probable, et ce, grâce au nombre d'occurrences relevées dans le corpus.

L'utilisation des catégories positionnelles permet d'étendre le contexte. Par exemple, dans le triplet *+nom :+artdem :+adja*, la catégorie *+adja* permet de savoir qu'ensuite il y a un nom.

Les couples (mot/catégorie positionnelle) sont utilisés pour déterminer lorsque les mots ont plusieurs catégories positionnelles possibles, celle qui semble être la plus pertinente (celle qui est très majoritairement utilisée est attribuée sauf contextes très particuliers). Mais donnons un exemple concret pour ces couples (mot/catégorie grammaticale positionnelle). Le mot *chien* peut être nom ou adjectif (exemple : *Je suis disposé à être rébarbatif, chien et insociable*). Dans le corpus, seul *chien* en tant que nom apparaît. Si seules les catégories sont prises en considération, pour la phrase *C'est un gentil chien.*, l'algorithme va choisir comme solution *gentil-nom* et *chien-adjectif postérieur* parce que la structure (nom-adjectif postérieur) est plus courante que la structure (adjectif antérieur-nom) en français. Mais en ajoutant les poids des couples (mot/catégorie), l'algorithme choisira la bonne solution, c'est-à-dire *gentil-adjectif antérieur* et *chien-nom*.

Une fois cette annotation terminée, les règles d'extraction combinées à l'ontologie de GEOLSemantics permettent de faire une désambiguïsation sémantique. Par exemple, dans la phrase « *le stade explose de joie* », l'explosion de joie ne sera pas annotée comme un acte de violence.

Au niveau de l'analyse morpho-syntaxique, la désambiguisation sémantique réalisée consiste à revoir le type des entités nommées selon le verbe employé. Lorsque l'entité nommée est sujet d'un verbe d'action, si c'est une EN de lieu, alors l'EN ne désigne pas le lieu mais l'organisme représentant le lieu. Par exemple, dans la phrase « *La France a décidé de...* », *France* désigne le gouvernement français.

Au niveau de l'extraction sémantique guidée par l'ontologie, nous interprétons les termes en fonction d'un domaine particulier, selon le client, ou bien selon les textes que nous traitons. Ainsi, nous interprétons les termes à la fois selon le domaine d'application (sécurité, droit, économie, etc.) et selon les contextes définis dans les règles. Par exemple, *inonder* est un terme a priori non ambigu. Mais dans les textes comme les tweets, il n'est pas possible d'exclure tout emploi métaphorique de ce terme. Bien au contraire. Par exemple, pour la phrase « *La rue est inondée de monde* », l'utilisation de règles sémantiques d'extraction fondées sur l'analyse syntaxique, permet de lever une grande partie des interprétations métaphoriques des termes ; même s'il est certain que certaines ne peuvent pas être repérées. Ici, le mot *inonder* est employé avec un complément d'objet direct, il s'agit donc d'une métaphore.

Dans la règle présentée en annexe A1 , on cherche, dans les phrases qui contiennent un pronom personnel sujet féminin à la troisième personne du singulier ($'trigger'$ : {$'token'$ : $tokens.PRONPERSSUJ_3PSF$}) s'il existe dans les mots précédents, un mot qui peut être antécédent de ce pronom. Dans ces antécédents, nous privilégions ceux en position de sujet car ce sont le plus souvent eux qui sont repris. La phrase « Marie arrive avec Paula, elle a amené des fleurs » en est une illustration.

Nous avons des règles pour chaque type de pronom. Si aucun antécédent-candidat est trouvé dans la phrase, nous regardons dans les phrases précédentes du paragraphe. Lorsque l'antécédent est trouvé, nous remplaçons le pronom par son antécédent (action $findAntecedent$).

**Exemple 3.3.1.** Règle pour la reconnaissance de concepts défini par :

$ID\ REL_NAME\ NODEID\ NAME\ POSITION\ REL_TYPE\ H_LEMMAS\ H_CATS\ H_PROPS\ D_LEMMAS\ D_CATS\ D_PROPS\ L_LEMMAS\ L_CATS\ CONTEXT\ 0\ FireDisaster\ dept\ isBurned\ head\ GD\ \$vehicles\ annpprod, nom * feu, flamme * * en * _$

Dans la règle de l'exemple 3.3.1, nous détectons un concept $FireDisaster$ lorsque le texte contient une relation telle que $voitureenfeu$ par exemple. Et nous en profitons pour indiquer l'objet qui brûle grâce à la propriété $isBurned$.

**Exemple 3.3.2.** Règle déterminant un nom commun ou un nom propre

115 $FireDisaster\ head\ isBurned\ dept\ VC\ incendier, bruler\ ***\ nom, np\ -loc*\$NoPrep$

_

Dans la règle de l'exemple 3.3.2, l'objet qui brûle n'appartient pas à une liste mais sera l'élément correspondant à un nom commun ou bien à un nom propre.

**Exemple 3.3.3.** Règle avec contexte

76 $FireDisaster\ head\ event-type[incendie]\ dept\ VC\ prendre\ **\ feu\ ****\ _$

134 $FireDisaster\ head\ isBurned\ dept\ VC\ prendre\ **\ \$vehicles\ ****\ 76$

Dans la règle de l'exemple 3.3.3, nous définissons un contexte qui est la relation verbe-complément prendre feu. Puis, nous précisons ce qui brûle (*isBurned*) grâce à la relation $verbe-complementprendre-listedevehicules$.

### 3.3.2 Désambiguïsation géographique

Un lieu peut être exprimé de plusieurs façons ; il peut être :

- simple : par exemple, *je vais à Paris* ;
- composé : *je suis au croisement de la rue A et de la rue B* ;
- flou : *je suis aux alentours de Paris.*

Afin de retrouver les coordonnées du lieu exact, nous avons décidé de faire deux modules pour gérer les lieux qui sont comparés/confrontés à une base de données de lieux indexée avec Elasticsearch (voir section 6.2.2).

Comme chaque requête renvoie plusieurs lieux potentiels, nous allons nous appuyer sur les modules suivants pour choisir le lieu exact :

- **selectGeoCoords** : ce module prend en argument une liste de lieux qui est le résultat du module POSTgal (ce module est détaillé au chapitre 7.3.3). Il permet de lever l'ambiguïté d'un lieu en se fondant sur des critères linguistiques ou sémantiques afin de sélectionner le résultat le plus pertinent parmi les résultats donnés par Elasticsearch (voir section 6.2.2) afin d'obtenir :

- des coordonnées (longitude, latitude) lorsqu'on a un nom de lieu précis (par exemple, *Tour Montparnasse*) ;
- des coordonnées les plus précises possible lorsque plusieurs noms de lieux candidats sont renvoyés comme cela pourrait être le cas par exemple avec le terme *République* (place), *République* (arrêt de bus), *République* (arrêt de métro).

— **compareGeoCoords** : ce module prend en argument une liste de lieux avec leurs coordonnées. Ce module a pour objectif de retourner les coordonnées les plus pertinentes par les étapes suivantes :

- comparer les coordonnées afin d'en sélectionner les coordonnées les plus précises
  - selon la relation exprimant la notion de contenant/contenu entre les lieux. Par exemple, la phrase « je suis à place de la République à Paris » : *Place de la République* est contenue dans *Paris*, donc il faut renvoyer *Place de la République* ;
  - en calculant les intersections entre les lieux : l'intersection va nous permettre de détecter la relation entre les lieux et nous donne une indication si on renvoie les deux lieux ou leur intersection ; cette notion est détaillée dans le chapitre 6 ;
- désambiguïser les lieux multiples juxtaposés, par exemple, *Paris, Texas* ;
- désambiguïser les lieux multiples avec préposition, comme par exemple, *entre rue Jean Moulin et boulevard Gambetta*) ;

— sélectionner le lieu selon des critères liés au vocabulaire :

- si l'annonceur de lieu est présent dans le nom de lieu. Par exemple, « *Je suis à l'aéroport de Nice* » implique qu'il ne faut pas renvoyer les coordonnées de la ville mais celle de l'aéroport ;
- si l'annonceur n'est pas présent, il faut faire une liste d'annonceurs prédéfinie et les organiser par ordre de priorité ("ville, commune, avenue", ...). Par exemple, « *Je suis à Nice* », en l'absence d'annonceur, on renvoie les coordonnées de la ville de Nice.

Afin de mieux étudier les lieux composés, nous avons effectué une première expérimentation. Pour cela, nous avons tout d'abord récupéré deux corpus de tweets avec certaines caractéristiques :

- corpus 1 (plusieurs prépositions dans le même Tweet) ;
  - répertorier une liste de prépositions ;
  - lancer une requête avec minimum deux prépositions dans les tweets ;
  - garder les tweets avec des lieux après les prépositions ;
  - garder les tweets avec le pattern de préposition le plus commun (30 occurrences minimums) ;
- corpus 2 (plusieurs lieux dans le même tweet) ;
  - lancer une requête avec minimum deux lieux dans les tweets ;
  - récupérer les patterns de prépositions ;
  - garder les tweets avec le pattern de préposition le plus commun (30 occurrences minimums).

Afin de définir les relations entre prépositions, nous avons constitué un premier corpus de prépositions. Nous avons ensuite récupéré des tweets en utilisant le corpus des prépositions issues de la première étape. Ceci nous a permis, dans un premier temps, d'améliorer le corpus des prépositions de lieux, mais aussi, en utilisant au moins deux prépositions dans une requête, de créer un corpus de tweets signalant au moins deux lieux. Nous en avons déduit un certain nombre de relations entre lieux que le tableau 3.1 résume.

Nous pouvons constater, après cette expérimentation, que des règles de détection peuvent exister indépendamment des prépositions. Par exemple, la relation d'inclusion pourrait se suffire à elle-même dans certains cas compte-tenu que 60% des lieux peuvent être résolus avec cette règle. Cette étude a révélé que, dans la plupart des cas, si un lieu est inclus dans un autre, c'est le lieu le plus petit qui est signalé dans le tweet. Le premier lieu cité est souvent une indication du lieu le plus précis, en français. Par exemple, $Paris, Texas$.

Dans le reste des cas (quand il n'y a pas inclusion), les prépositions utilisées aident à définir le lieu. Nous avons mis les prépositions les plus utilisées dans un fichier de règles (pre-

| **Type de relation entre lieux** | **Nombre de cas (% des tweets)** | **Exemple** |
|---|---|---|
| Inclusion (Lieu1 est inclus dans Lieu2) | 62 | au quartier nord de Marseille |
| Croisement (Lieu1 traverse Lieu2) | 3 | au carrefour de la D24 et de la D112 |
| Zonage (Zone autour des deux lieux) | 12 | entre boulevard Wilfrid-Hamel et rue Einstein |
| Autres | 23 | dans l'angle de la rue derrière l'hôtel de ville d'Amiens |

TABLEAU 3.1 – Relation entre les lieux

positions.rules)3.1 indiquant ce qu'il faut faire pour chaque combinaison de prépositions (combinaison, intersection, renvoyer un des lieux, etc.).

L'algorithme 1 a été développé suite à cette première étude :

**Algorithme 1 :** Algorithme de regroupement des lieux

**Data :** $zoneL1$ : zone autour du lieu L1 ; $zoneL2$ : zone autour d'un autre lieu
**Result :** $listeDesZonesDeLieux$, liste des zones de lieux
  $listeDesZonesDeLieux \leftarrow zoneL1, zoneL2$
  **if** $zoneL1 \bigcap zoneL2 \neq \varnothing$ **then**
    $listeDesZonesDeLieux \leftarrow L1 \bigcap L2$
  **else if** les prépositions utilisées dans $L1 \bigcap L2$ sont dans $prepositions.rules$ **then**
    $listeDesZonesDeLieux \leftarrow$ la zone de lieu défini dans $prepositions.rules$
  **end if**
  renvoyer $listeDesZonesDeLieux$

On a vu que la relation déterminée entre les lieux (inclusion, intersection, aucune relation) est importante. Afin de finaliser la liste des $prepositions.rules$, nous avons dû faire une étude plus complète en collectant des tweets sur :

— Paris-Place de la république (inclusion),

— Paris-Gentilly (intersection),

— Paris-Nice (aucune relation).

La procédure de constitution de ce nouveau corpus de tweets se fait comme suit :

— former la requête avec les lieux dans n'importe quel ordre et toutes les possibilités de prépositions ; par exemple, *de Paris à Nice*, *de Nice à Paris*, *à Nice de Paris*, *de Paris à Place de la République*, *de Paris ...*

— récupérer au moins 100 tweets pour chaque requête.

Signalons que le nombre total de requêtes généré automatiquement est de 37 454 mais seulement 296 d'entre elles ont effectivement renvoyé des tweets selon la répartition suivante :

— 55 requêtes avec des lieux inclus (Paris-Place de la République) ;

— 50 requêtes avec des lieux en intersection (Paris-Gentilly) ;

— 191 requêtes avec lieux non connectés (Paris-Nice)..

En regardant de plus près les trois types de requêtes nous pouvons remarquer que :

— si le lieu est inclus, c'est le lieu ciblé. Il faudra toutefois le nuancer grâce à la préposition qui le précède (vers, à côté de, ...) ;

— si une intersection est possible (souvent, un des lieux est cité comme s'il était inclus dans l'autre, par exemple *à Gentilly, Paris* laisse supposer que Gentilly est inclus dans Paris or il apparaît des coordonnées appartenant aux deux communes qui elles sont bien distinctes), dans un premier temps, nous allons chercher si une préposition nous permet de décider lequel des deux lieux retenir sinon on renvoie l'intersection elle-même ;

— si les deux lieux sont séparés : souvent, ce n'est pas un lieu qui est ciblé mais c'est le trajet entre *lieu1* et *lieu2*. Ce sont alors les prépositions qui permettent de distinguer entre deux lieux distincts et un trajet.

Afin d'obtenir un maximum de relations entre deux lieux, mais aussi composer notre liste des règles linguistiques gérées par les prépositions, nous avons procédé comme suit :

— faire une liste de prépositions,

— récupérer un corpus de tweets avec la liste des prépositions,

— faire une annotation automatique des lieux en utilisant le module GAL (cf. figure 1.5),

— faire le pourcentage de lieux réels pour chaque préposition (si présente, combien de lieux vont suivre),

— filtrer les lieux qui suivent les prépositions,

— trier les lieux par catégorie (nom propre, lieu défini, zone géographique, etc.),

— refaire un corpus pour chaque catégorie de lieux,

— compléter la liste des prépositions avec les prépositions qui apparaissent dans le nouveau corpus,

— faire un corpus avec toutes les combinaisons possibles des prépositions,

— récupérer semi-automatiquement la liste des prépositions qui relient deux lieux.

Après ce travail, nous avons séparé les prépositions et celles plus spécifiques liées à des lieux dans différentes catégories selon la classification suivante :

— **lieu précis** : à, chez, sur, dans, au beau milieu de, au centre de, au cœur de, au milieu de, le long de, le plus proche de, proche de, Lieu flou proche de, à côté de, à proximité de, à la hauteur de, aux alentours de, autour de, aux environs de, auprès de, du côté de, vers, non loin de, près de, contre, proche de, sous, l'autre bout ;

— **lieu proche avec indication de direction** : à droite de, à gauche de, de l'autre côté de, derrière, devant, en face de , face à, en arrière de, à l'arrière de, au dessous de, au dessus de, en deçà de, en amont de, en aval de, au large de

— Une intersection : au carrefour de .... Et de, au coin de .... et de, au croisement de X et de, à l'intersection de .... et de ;

— **lieu situé par rapport à deux lieux** : de.*à, de.*Jusqu'à, depuis.*jusqu'à, entre.*et ;

— lieu par rapport à un polygone : à l'extérieur de, à l'intérieur de, à l'entrée de, à la sortie de, aux portes de, hors de, en dedans de, en dehors de, faire le tour du ;

— **lieu flou précisé par un point cardinal** : à l'Est de, à l'ouest de, au Nord de, au Sud de, etc. ;

— **lieu flou ambigu** : au nord de, au sud de, à l'est de..., en bas de, en haut de, au-dessous de, au-dessus de ;

— **lieu indiquant un but ou une provenance** : à destination de, en provenance de, en direction de, après, avant, à travers, au travers de, vers le bas de, vers le haut de, jusqu'à ;

— **point éloigné flou** : expression loin de.

Nous avons ensuite utilisé ces résultats afin de créer le fichier (voir listing 3.1) dans lequel est précisé la façon dont doivent être affichés les lieux selon les prépositions.

Ce fichier est utilisé à l'intérieur du système GEOLsig afin de désambiguïser les tweets possédant plusieurs lieux en renvoyant les coordonnées et le nom à afficher soit par combinaison entre deux lieux, soit les deux lieux, soit un seul lieu comme détaillé dans l'algorithme 1.

L'algorithme cherche le *pattern* des lieux qu'il a en argument dans le fichier *prépositions.rules*, et renvoie le résultat selon la commande signalée dans le fichier. Les commandes sont :

— combinaison : il combine les deux lieux et renvoi le résultat. Exemple *lieu1 à destination de *lieu2

— lieu1 : renvoi le lieu 1. Exemple : *lieu1 aux portes de *lieu2

— lieu2 : renvoi le lieu 1. Exemple : hors de *lieu1 à *lieu2

Listing 3.1 – Échantillon du fichier prepositions.rules

```
*lieu1 aux portes de *lieu2      lieu1
*lieu1 en direction de *lieu2    combinaison
de *lieu1 vers *lieu2    combinaison
*lieu1 loin de *lieu2    lieu1
...
```

### 3.3.3 Phonétisation

Nous avons déjà énoncé que les tweets sont rédigés dans une langue peu soignée. Il se trouve aussi que les rédacteurs de tweets ont tendance à faire des fautes d'orthographe en

écrivant les termes de manière phonétique.

Les erreurs les plus courantes surviennent en fin de mot sur les déclinaisons qui se prononcent de la même manière et qui sont confondues par le rédacteur comme, par exemple, « *avait* » au lieu de « *avaient* ». De même, les mots courts avec accent sont souvent écrits sans celui-ci, comme « *a* » au lieu de « *à* ». De plus, les personnes écrivent parfois avec des fautes d'orthographe dans les mots eux-mêmes. Pour limiter les ambiguïtés générées par ces rédactions, nous avons développé deux systèmes de phonétisation à savoir :

1. phonétisation des fins identiques ; par exemple, « *ais, ait, aient, ai, ...* » renvoient aux mêmes possibilités ;
2. phonétisation des mots lorsqu'ils sont toujours inconnus après la première étape ; par exemple, « *kalme* ». sera transformé dans un premier temps en kalmais, kalmait, ... Aucun mot ne sera reconnu, mais la phonétisation complète va permettre de trouver « *calme* ».

Le phonétiseur réalisé par GEOLSemantics est utilisé à deux niveaux à savoir (i) sur le dictionnaire pour phonétiser les mots de ce dernier et (ii) sur chaque mot du Tweet pour pouvoir aligner le texte avec les mots du dictionnaire phonétisé.

Nous avons pris aussi en considération le fait que certaines erreurs sont effectuées par convention entre les internautes. Certains mots sont entrés dans le domaine de l'argot et utilisés en masse dans les réseaux sociaux. Pour cette raison, nous avons décidé de créer un dictionnaire de mots argotiques selon la procédure suivante :

— récupérer un corpus de tweets avec comme mots-clés les mots argotiques les plus communs ;[7]

— analyser ce corpus par les outils GEOLSemantics ;

— récupérer les mots annotés comme inconnus ayant un nombre d'occurrences supérieur à 100 ;

— ajouter les mots dans le dictionnaire après une vérification manuelle de l'existence de ce mot.

7. `http://www.dico2rue.com/`

## 3.4 Conclusion

Nous avons présenté dans ce chapitre les différentes problématiques qui accompagnent le traitement linguistique d'un texte. Nous avons explicité le panel de ce qui existait à GEOLSemantics. Nous en avons déduit un certain nombre d'amélioration à envisager notamment dans la désambiguïsation géographique.

Mais pour pousser plus loin notre investigation, il est nécessaire d'avoir de bons corpus de termes alignés pour faciliter la désambiguïsation. Dans le prochain chapitre, nous allons nous atteler à cette tâche en mettant en place une procédure de constitution d'un corpus à partir de tweets collectés sur une zone géographique précise, ainsi que l'alignement du corpus qui va être utilisé afin de réaliser ce projet.

# Chapitre 4

# Corpus d'alerte

## Sommaire


4.1 Méthodes de constitution du corpus . . . . . 82
4.2 Corpus de recherche Twitter . . . . . 88
4.3 Interfaces de constitution du corpus . . . . . 90
4.4 Exemple de création de corpus . . . . . 97
4.5 Corpus d'expérimentation . . . . . 102
4.6 Premières constatations . . . . . 105
4.7 Conclusion . . . . . 108


Contrairement à d'autres réseaux sociaux plus populaires, Twitter donne un accès limité à ses données. En contrepartie, Twitter peut aider les chercheurs à comprendre l'opinion publique, mesurer l'intérêt suscité par certains types de sujets ou détecter en temps réel certains événements imprévisibles.

Cependant, les spécificités de Twitter (textes courts, utilisation de l'argot, abréviations, hashtags, images et vidéos, très grand volume de données) rendent toutes les tâches de détection automatique très difficiles sur des jeux de données brutes de tweets. D'autre part, un manque de corpus de tweets accessibles au public rend les expérimentations et les comparaisons difficiles entre travaux de recherche comme nous allons le voir dans la section 4.4.

Nous avons vu précédemment que pour les besoins du projet *SafeCity*, il a fallu constituer un corpus représentatif relatif à un nombre d'événements cités dans le cahier des charges du projet. Le corpus de tweets va nous aider, dans un premier temps, à décider quels concepts doivent être ajoutés à notre ontologie, et, dans un deuxième temps à écrire des règles d'extraction les plus générales possible, favorisant ainsi la portabilité de notre solution dans d'autres contextes.

Le but de la constitution de nos corpus est de servir à l'analyse du comportement linguistique des personnes qui signalent des événements. Il s'agit donc de mettre en exergue les écarts existants entre l'écriture des messages et le langage naturel quant à la manière dont les lieux sont désignés, le vocabulaire utilisé, ainsi que les fautes et expressions contractées les plus courantes. Cette analyse a été complétée par des corpus générés pour un autre projet portant sur les relations entre influenceurs et influencés. Cela nous a permis de constater que cet autre domaine, plutôt pratiqué par des jeunes pour ce qui concerne les personnes influençables, possède des caractéristiques de langues beaucoup plus éloignées de la langue naturelle et avec aussi un vocabulaire particulier lié au thème.

Même si la finalité de détection d'événements envisagée dans le projet *SafeCity* ne nécessite pas la prise en compte d'une langue très « abîmée », le développement d'un analyseur de tweets doit pouvoir être robuste même pour ce type de langue.

## 4.1 Méthodes de constitution du corpus

Pour la constitution de corpus de tests, nous avons considéré trois voies possibles :

1. **par compétition** : la première idée était de participer à des campagnes d'évaluation pour obtenir un corpus bien défini et évaluer nos outils, comme dans "*Named Entity Recognition in Tweets* - CAp 2017" [Lopez *et al.*, 2017]. Malheureusement, le corpus mis à disposition lors de cette compétition n'était pas représentatif des événements à considérer dans notre projet puisque c'est un corpus de tweets en français avec des entités nommées et ne comportant pas de métadonnées spécifiques notamment de géolocalisation, indispensables pour le projet *SafeCity*. D'autre part, chercher la

bonne campagne d'évaluation avec le bon corpus aurait été une tâche inutilement chronophage ;

2. **par *scrapping*** : la deuxième approche est une approche algorithmique consistant à concevoir ou utiliser un programme interrogeant l'interface web de Twitter afin de récupérer la page web de la réponse à notre requête et, ensuite, extraire de cette page HTML les tweets en relation avec notre requête et les métadonnées disponibles, en utilisant l'outil gratuit Twint [1].

3. **par l'utilisation de l'API Twitter** : toujours en restant dans l'approche algorithmique, nous avons fait un programme qui va interroger l'API officielle de Twitter pour récupérer les tweets et un nombre considérable de métadonnées indisponibles sur l'interface Twitter. Avec la version gratuite, nous ne pouvons récupérer que 1% des tweets à un instant t (par l'API Stream [2]) ou filtrer les tweets selon des mots-clés, les coordonnées de géolocalisation et l'identifiant (ID) de l'utilisateur ou la langue de rédaction du tweet. Avec l'API REST [3], des tweets spécifiques peuvent être extraits, notamment les tweets d'un utilisateur en particulier (les 3200 plus récents), mais aussi la liste des *followers* d'un certain utilisateur, ou les tweets correspondant à une requête par bounding box c'est à dire un espace géographique donné. Dans ce cas, seuls les tweets des utilisateurs qui autorisent l'application à se connecter à leurs coordonnées GPS sont récupérables. Notons que la version gratuite met fin à la requ ête toutes les quinze minutes. Dans ce laps de temps, le nombre de requêtes autorisées va de 180 à 450 demandes selon le type d'authentification [4] Steinert-Threlkeld [2018]. Il faut alors attendre 15 minutes pour relancer une autre requête Steinert-Threlkeld [2018].

Malgré la stabilité et la robustesse de l'API Twitter, compte-tenu des restrictions très fortes quant à son utilisation en version gratuite, nous avons décidé d'utiliser le *scrapping* par Twint car la quantité de tweets recupérés est bien supérieure et le texte collecté nous permet de constituer un corpus bien plus complet et pertinent à soumettre à la chaîne de

1. Twitter Intelligence Tool, https ://github.com/twintproject/twint/wiki
2. https ://developer.twitter.com/en/docs/tutorials/consuming-streaming-data
3. https ://developer.twitter.com/en/docs/api-reference-index
4. https ://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets

traitement GEOL (cf. figure 5.3).

Nous avons néanmoins utilisé l'API Twitter afin d'extraire les tweets par délimitation rectangulaire d'une zone géographique, dite *bounding box*, sur la zone de Nice, ainsi que pour récupérer des métadonnées que Twint ne peut pas extraire. Le moteur de notre interface de constitution de corpus 4.3 est donc composé à la fois de l'API Twitter et de Twint.

Il est bien connu de tous maintenant que la gestion des données personnelles reste un sujet très sensible. Nous allons dans ce qui suit présenter les restrictions selon Twitter et selon la RGPD [5] concernant la récupération, le traitement et le partage d'un corpus de tweets dans le domaine de la recherche.

### 4.1.1 Le *crawling* selon Twitter

Conformément à l'article 8 des conditions d'utilisation de Twitter, traitant des restrictions sur le contenu et l'utilisation des services [6], dans le cadre de l'accès gratuit, on ne peut pas accéder ou rechercher ou tenter d'accéder services par tout moyen (automatisé ou autre) autrement que par le biais des interfaces publiées actuellement disponibles et fournies par Twitter (et uniquement conformément aux conditions applicables, sauf si on a été spécifiquement autorisé à le faire dans un accord séparé avec Twitter.

L'analyse des services est autorisée si elle est effectuée conformément aux dispositions du fichier *robots.txt*. Cependant, *scrapper* les services sans le consentement préalable de Twitter est expressément interdit (voir conditions d'utilisation).

Les *crawlers* devraient idéalement suivre le fichier *robots.txt* pour tout site web. Il contient des règles spécifiques pour un bon comportement, telles que la fréquence de *scrapping*, les pages qui permettent le *crawling* et les pages interdites. Le protocole d'exclusion des robots est une ressource de format texte qui peut être placé à la racine d'un site web. Il contient une liste des ressources du site qui ne sont pas censées être indexées par les robots

5. `https://www.cnil.fr/fr/comprendre-le-rgpd`
6. voir règlement, version 10

d'indexation des moteurs de recherche. Par convention, les robots consultent *robots.txt* avant d'indexer un site web[7].

### 4.1.2 Le *crawling* selon la RGPD

Essentiellement, il est illégal de *scrapper* les données personnelles d'un résident de l'Union européenne (UE), sauf en cas de consentement. En vertu de la RGPD, les informations suivantes sont toutes considérées comme des informations personnellement identifiables (dit "*personally identifiable information*" (PII)) :

— nom,
— adresse physique,
— adresse électronique,
— numéro de téléphone,
— détails de la carte de crédit,
— coordonnées bancaires,
— adresse IP,
— date de naissance,
— informations sur l'emploi,
— numéro de sécurité sociale,
— informations médicales,
— enregistrements vidéo / audio.

Pour récupérer ces types de données personnelles, on doit avoir une raison légale de le faire. En vertu de la RGPD, il existe plusieurs voies légales :

— par consentement : la personne concernée (personne dont nous avons les données) a consenti à ce que nous ayons ses données ;
— par contrat : les données personnelles sont nécessaires à l'exécution d'un contrat avec la personne concernée ;

7. cf. twitter.com/robots et les bonnes pratiques de crawling

— par conformité : nécessaire au respect d'une obligation légale ;

— par intérêt vital, intérêt public ou autorité officielle : généralement applicable uniquement aux organismes gérés par l'État lorsque l'accès aux données personnelles est dans l'intérêt public ;

— par intérêt légitime : nécessaire à nos intérêts légitimes.

En ce qui concerne le *scrapping*, les raisons juridiques les plus courantes sont l'intérêt légitime et le consentement.

Malheureusement, l'intérêt légitime ne peut être soutenu de manière fiable que si on est un gouvernement, un organisme chargé de l'application des lois, etc. ; et l'obtention du consentement implique que le résident de l'UE accepte explicitement un contrat qui permette à l'entreprise de crawler et de stocker ses données personnelles `https://blog.scrapinghub.com/web-scraping-gdpr-compliance-guide`.

### 4.1.3 Le *crawling* dans le domaine de la recherche

L'utilisation des sites web, des SMS, des APIs, des notifications par e-mail, des applications, des boutons, des intégrations, des publicités et des autres services couverts par Twitter est régie par des conditions générales de service et une politique de confidentialité ou contrat de développeur [8].

Dans le contexte de la recherche, il est clairement dit que nous ne pouvons pas publier de données provenant de Twitter en tant que données ouvertes. La politique des développeurs de Twitter impose un certain nombre de restrictions à l'utilisation de leur API et aux données à obtenir. Notamment, dans les restrictions sur l'utilisation des matériaux sous licence, nous ne pouvons pas utiliser les données géographiques de la plate-forme sauf pour identifier l'emplacement à partir duquel un tweet a été rédigé et non à d'autres fins.

Selon la section F.2 du contrat et de la politique du développeur de Twitter, on accepte de ne rien stocker, à l'exception de l'ID d'un tweet. On ne peut pas stocker le message, ni les métadonnées ou quoi que ce soit d'autre sur l'utilisateur. Cependant, même avec l'ID, il

8. developer.twitter.com/.../agreement-and-policy

existe des restrictions sur le nombre de tweets par utilisateur et par jour. De toute évidence, ces termes sont conçus pour le seul avantage de Twitter. Ils les aident à conserver autant de valeurs que possible tout en fonctionnant comme une plate-forme. Twitter applique rarement ces conditions, même pour des services qui les enfreignent clairement. Après avoir rédigé un accord juridique, ils choisissent de ne pas faire de surveillance active car au final, cela ne vaut pas la peine de le faire. D'autre part, cela constitue une bonne publicité de montrer comment on peut utiliser les données de Twitter. Vraisemblablement, ils interviendraient s'il y avait des violations importantes à grande échelle...

Lors de la récupération des données, il peut ne pas être possible d'obtenir le consentement de tous les participants, simplement en raison du volume de tweets récupérés. Notons aussi que des problèmes éthiques peuvent survenir si on décide de reproduire des tweets dans une publication académique. Certains tweets doivent être traités avec soin, en particulier lorsqu'ils sont liés à des sujets sensibles.

Le partage d'ensemble de données est interdit dans les conditions d'utilisation de l'API de Twitter mais nous avons vu que nous pouvons partager les numéros d'identification des tweets (ID), associés à chaque tweet, qui peuvent être utilisés par d'autres chercheurs pour obtenir des ensembles de données Twitter. Si, pour une raison quelconque, il n'est pas possible de partager des identifiants de tweet, le partage des mots-clés et du temps de récupération des données peut permettre aux chercheurs d'obtenir un ensemble de données similaire. Il peut également y avoir des exigences spécifiques pour la production de tweets dans une publication ; il faut se référer aux directives de Twitter (cf. publication de tweets comme données ouvertes).

### 4.1.4 Contexte de l'étude et limitations de Twitter

Les limitations de Twitter se sont pas des moindres quand il s'agit de réaliser des expérimentations à des fins d'analyse de contenus sur des réseaux sociaux. Dans notre cas, l'optimisation des algorithmes est incontournable pour arriver à détecter au plus tôt un événement qui devrait générer une alerte. Pour outrepasser les limites fixées par Twitter afin de nous permettre d'effectuer des expérimentations acceptables, tout en respectant

ses conditions d'utilisation, les voies suivantes ont été envisagées :

— ne proposer que l'identifiant du tweet accompagné des triplets de l'événement et des mots-clés, sans mettre le texte du tweet. L'identifiant rend reproductible tout expérience par des chercheurs car cela donne indirectement le corpus (risque zéro par rapport à Twitter et la RGPD). Cette solution a été aussi adoptée par [Mazoyer *et al.*, 2020] ;

— mettre le texte du tweet tout en enlevant toutes données personnelles mentionnées dans le tweet. Dans ce cas, nous allons donner directement le corpus avec les informations personnelles anonymisées. Tout chercheur peut récupérer les tweets en utilisant l'identifiant du tweet (cela est plus risqué s'il y a un partage en masse mais moins risqué qu'un partage mot par mot) ;

— demander une autorisation de Twitter mais, même avec cela, il faut être conforme à la RGPD.

À la lumière de ces précisions, nous avons décidé de créer plusieurs corpus de développement et de tests selon différents critères détaillés dans ce qui suit. Nous allons aussi présenter un état de l'art des corpus de recherche français orientées détection d'événements et d'entités nommées.

Au vu de cela, nous avons décidé de partager notre corpus d'expérience du chapitre 4.2 avec la communauté de chercheurs. Cependant, afin de rester dans la légalité, nous allons fournir que les identifiants des tweets et les annotations que nous avons réalisées. Grâce aux identifiants du tweet, les utilisateurs peuvent récupérer le tweet, du moment qu'il n'est pas supprimé, dans le respect de la RGPD, de Twitter et du droit à l'oubli des twittos.

## 4.2 Corpus de recherche Twitter

Rappelons que Twitter, site de *micro blogging* existant depuis le 15 juillet 2006, est souvent un terrain de jeu pour les chercheurs car il est plus souple sur l'utilisation des données que les autres plates-formes, allant même jusqu'à proposer une plate-forme pour la récupération des tweets. Malgré cela, il existe peu de corpus mis à disposition pour la communauté

de chercheurs permettant de comparer les systèmes d'analyse de tweets et donc la pertinence des résultats. Nous allons présenter maintenant quelques corpus de tweets constitués pour la recherche concernant la notion d'événement en français et en anglais, thème de notre étude, même s'il en existe d'autres pour d'autres sujets d'étude (par exemple, le corpus de *code switching* avec des étiquetages de sentiments proposé par Vilares *et al.* [2015] ou le corpus des messages liés à des crises qui ont eu lieu entre 2013 et 2015 proposé par Imran *et al.* [2016]).

Mazoyer *et al.* [2020] présentent *Event2018*, un corpus annoté pour les tâches de détection d'événements, composé de 38 millions de tweets en français (retweets exclus) dont plus de 130 000 tweets annotés manuellement comme liés ou non à un événement donné. Les 257 événements retenus ont été sélectionnés à partir d'articles de presse pendant la période d'annotation (juillet à août 2018). Ils détaillent également les résultats de leurs expériences de détection d'événements sur cet ensemble de données et sur un autre ensemble de données de tweets accessible au public en anglais. Le corpus est disponible sur le site de l'INA. Conformément aux conditions d'utilisation de Twitter, ils ne fournissent que les identifiants de tweet et un script Python pour récupérer le texte intégral des tweets à partir de l'API Twitter [9].

McMinn *et al.* [2013] ont créé le plus grand corpus disponible sur la détection d'événements. Ils ont utilisé des méthodes de détection automatique d'événements et une méthode basée sur l'expansion des requêtes d'événements. Le corpus est composé de 120 millions de tweets dont plus de 100 000 sont annotés couvrant 506 événements.

La base de données SNOW [Papadopoulos *et al.*, 2014] est composé de 59 événements issus des gros titres du fil d'actualité RSS de la *BBC* et de *NewsWhip UK* publiés pendant la journée du 25 février 2014.

Le corpus *Signal-1M* est composé de deux ensembles de données à savoir un ensemble de données d'un million d'articles de presse de septembre 2015 [Corney *et al.*, 2016] et un ensemble de données de tweets liés à 100 articles sélectionnés au hasard [Suarez *et al.*, 2018]. Cette base de données contient environ 6000 tweets.

9. voir `github.com/ina-foss/twembeddings`

M. Mele et Crestani [Mele and Crestani, 2019] présentent une base de données d'événements qui contient 147 000 documents provenant de 9 chaînes d'information et publiés sur Twitter, RSS et sites d'informations de mars à juin 2016. 57 événements médiatiques ont été annotés sur une plate-forme de production de contenu pour étiqueter 4300 documents dont 744 tweets.

*Storyzy* est spécialiste dans la protection de l'image de marque par la détection du contenu suspect sur les sites Web pouvant potentiellement afficher les publicités des marques. Pour leur Hackaton 2018 [10] ils ont fourni un corpus contenant des textes de divers sites Web en anglais (4105 articles), en français (941 articles) ainsi que des transcriptions automatiques de vidéos Youtube en français (312 messages), concernant la vaccination. Les titres des articles partagent quelques particularités avec les tweets (comme la longueur). Il est à noter que l'utilisation du titre comme facteur discriminant a permis d'obtenir de bons résultats dans la détection de fausses nouvelles (ou *fake news*) [Guibon *et al.*, 2019].

Les corpus de *Storyzy* et *Event2018* sont les plus importants pour nous compte-tenu qu'ils sont en français, sur des contextes événementiels et qu'ils vont nous permettre d'améliorer notre solution en ajoutant la détection des *fake news*. Ils vont aussi nous donner la possibilité d'étendre notre solution sur d'autres types d'événements.

## 4.3 Interfaces de constitution du corpus

Afin de pouvoir réaliser des expérimentations plus exhaustives, nous avons ressenti la nécessité d'automatiser la constitution de corpus grâce à des interfaces paramétrables par des utilisateurs finaux.

L'interface de requêtage (cf. figure 4.1) permet à tout utilisateur de lancer une requête sur l'API Twitter ou l'outil Twint 4.1 selon les besoins, puis, de filtrer les tweets récupérés par une expression régulière ; l'interface affiche alors les tweets correspondants (cf. figure 4.2). L'utilisateur peut ensuite choisir les tweets qui lui semblent pertinents et les stoker dans une base de données relationnelles ou dans un document RDF [11] (structure des tweets).

10. `https://hackatal.github.io/2018/`
11. Resource Description Framework ; voir `https://www.w3.org/RDF/`

La page d'accueil de l'interface (cf. figure 4.1) a trois champs. Le champ obligatoire correspond à l'expression de la requête de l'utilisateur, en utilisant des opérateurs simples imposés par l'API Twitter [12] dont 'AND', 'OR', '()', etc., ce qui reste très général. Donc, pour affiner le corpus résultant de la requête, l'utilisateur peut utiliser le champs optionnel « Votre expression régulière ». Par exemple, l'utilisateur peut mettre Paris$ pour ne récupérer que les tweets qui finissent par Paris.

Une fois le corpus récupéré et avant de le stocker dans la base de données, les tweets qui ne correspondent pas à l'expression régulière seront ignorés.

L'interface, ensuite, visualise les tweets résultant de la requête (cf. figure 4.2) avec les métadonnées suivantes :

— pseudo de l'utilisateur,

— texte du tweet,

— nombre de personnes qui ont marqué qu'ils aimaient le tweet,

— nombre de retweets,

— date du tweet,

— emplacement de l'utilisateur au moment du tweet si disponible par géolocalisation sinon celui annoncé par l'utilisateur dans le tweet (traitement spécifique détaillé dans le paragraphe 4).

L'interface repose sur l'environnement de développement Web *Django* [13]. Elle est toujours en développement bien qu'elle permette déjà de constituer le corpus. Elle peut être améliorée afin d'éviter des post-traitements en relation avec la mise en forme des tweets. Nous allons devoir la rendre encore plus intuitive notamment afin d'afficher des statistiques telles que les utilisateurs qui contribuent le plus, ou les *hashtags* qui reviennent le plus dans les tweets récupérés à partir d'une requête donnée. Elle doit aussi permettre à l'utilisateur de choisir manuellement des tweets de différentes requêtes afin de les stocker séparément. Notons que les données personnelles sont traitées à la volée et ne sont pas stockées. Par

12. voir `https://developer.twitter.com/en/docs/tweets/search/guides/standard-operators.html`
13. voir `https://www.djangoproject.com/`

contre, d'autres données, nécessaires pour le recoupement d'informations, comme l'auteur du tweet sont *hashées*[14] et seul le résultat du hashage est stocké. Cela nous permet de savoir si plusieurs tweets appartiennent à un même auteur ou si une personne est citée dans plusieurs tweets et, ainsi, suivre son parcours sans l'identifier explicitement ni stocker son nom.

Cependant, une problématique s'impose dans le cas des noms d'une même personne avec différentes orthographes. Ce cas est traité chez GEOLSemantics avec un phonétiseur (voir 4). Si deux noms ont une en commun, c'est qu'il s'agit peut-être de la même personne. Par exemple, *Sofie Do* vs. *Sophie Do*.

Puisque la RGPD interdit de façon générale de stocker les noms des personnes, afin de les comparer par le phonétiseur, nous allons stocker les hashs comme le montre l'algorithme 3.

**Algorithme 2 :** Algorithme du phonétiseur

**Data :** *nom*, chaîne de caractères du nom propre à phonétiser
**Result :** *dicoPhone*, une liste des différentes phonétisations possibles d'un nom

**Algorithme 3 :** Algorithme de repérage des déclinaisons de noms hashés

**Data :** *nom*, nom propre *dicoPhone*, sortie du phonétiseur en prenant *nom* comme argument, *bdd* base de données qui stocke les noms hashés
**Result :** *hashNom*, le nom hashé par la bibliothèque Python

```
1 begin
2     for chaque phonetisation dans dicoPhone do
3         if phonetisation hashée est dans bdd then
4             stocker le hash de la phonetisation
```

**Structure des tweets**[15] Dans le fichier de sortie XML, chaque tweet est stocké entre les balises *<tweet></tweet>*. Pour chaque tweet, les balises suivantes sont utilisées pour obtenir de l'information :

```
<text> </text>
<contributor> </contributor>
```

14. https://www.programiz.com/python-programming/methods/built-in/hash
15. voir developer.twitter.com

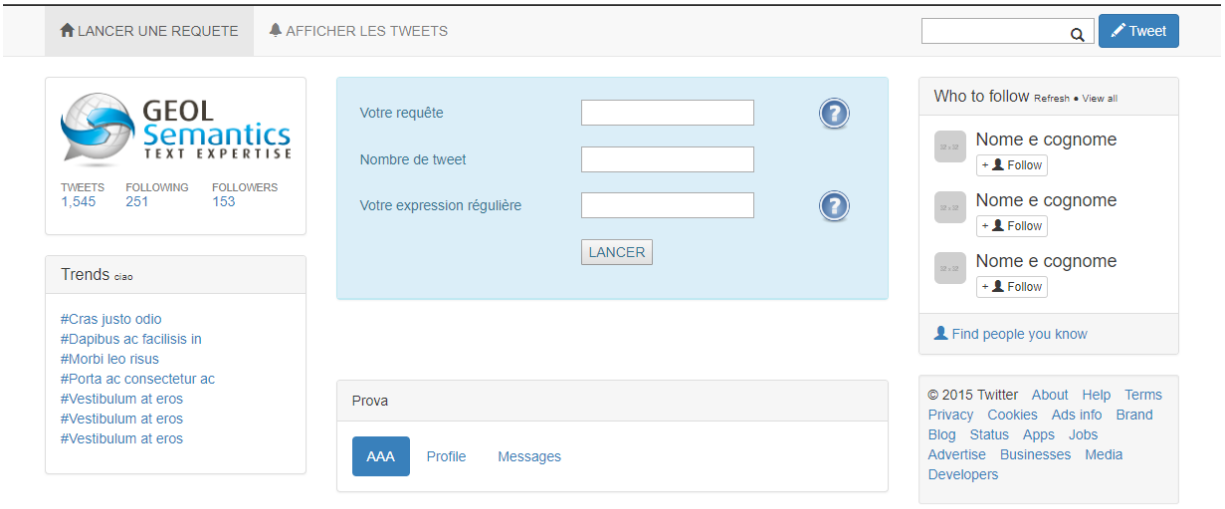



FIGURE 4.1 – Interface : lancement d'une requête

```
<time> </time>
<geo>
        <Coordinates> </Coordinates>
        <type_coordinates> </type_coordinates>
        <bounding_box> </bounding_box>
        <type_bounding_box> </type_bounding_box>
        <full_name> </full_name>
        <place_type> </place_type>
        <country> </country>
        <author_location> </author_location>
</geo>
<myQuery> </myQuery>
<lang> </lang>
<fav> </fav>
<retweet> </retweet>
<user_mentions> </user_mentions>
<hashtags> </hashtags>
<urls> </urls>
```

— text : le contenu du tweet avec toutes les citations, les hashtags et les urls ;

— contributor : le twittos [16] c'est-à-dire l'utilisateur de Twitter qui a publié le tweet

16. voir `https://fr.wiktionary.org/wiki/twitto`




FIGURE 4.2 – Interface : résultat de la requête

hashé ;

— time : la date et l'heure exacte du tweet ;

— geo : regroupement de toutes les données géographiques disponibles du tweet ;

— coordinates : les longitude et latitude exactes de l'emplacement géographique du twittos, telles que signalées par l'application ;

— type coordinates : le type des coordonnées (polygone, point, cercle, multipoints) ;

---

**Exemple 4.3.1.**

```
"coordinates": {
  "coordinates": [
    -75.14310264,
    40.05701649
```

```
  ],
  "type": "Point"
}
```

---

— bounding_box : les longitudes et latitudes exactes des quatre points de la zone géographique du twittos, telles que signalées par l'application ;

— type bounding_box : le type de la zone géographique définie dans la *bounding box* ;

— full_name : le nom de la ville du twittos au moment du tweet ;

— place_type : le type de la ville (par exemple : city, admin, ...) ;

— country : le nom du pays du twittos au moment du tweet ;

---

**Exemple 4.3.2.**

```
"place": {
  "attributes": {},
  "bounding_box": {
    "coordinates": [
      [
        [
          -77.119759,
          38.791645
        ],
        [
          -76.909393,
          38.791645
        ],
        [
          -76.909393,
          38.995548
        ],
        [
```

```
            -77.119759,
            38.995548
          ]
        ]
      ],
      "type": "Polygon"
    },
    "country": "United States",
    "country_code": "US",
    "full_name": "Washington, DC",
    "id": "01fbe706f872cb32",
    "name": "Washington",
    "place_type": "city",
    "url": "http://api.twitter.com/1/geo/id/0172cb32.json"
}
```


---

- author_location : le nom de la zone géographique du twittos tel qu'il veut la déclarer ;
- myQuery : la requête qui a permis l'obtention du tweet ;
- lang : la langue du tweet ;
- fav : le nombre de twittos qui ont ajouté le tweet dans leurs favoris ;
- retweet : le nombre de twittos qui ont retwitté le tweet.

  *fav* et *retweet* sont des éléments qui vont nous aider à calculer la portée du tweet et donc décider de sa crédibilité ;
- user_mentions : les autre profils cités dans le tweet hashé ;
- hashtags : les hashtags cités dans les tweets qui vont nous permettre de retrouver tous les tweets qui parlent du même sujet ;
- urls : toutes les urls mentionnées dans les tweets.

Un abonné de Twitter peut donner l'autorisation de stocker ses données de localisation. Dans ce cas, toute application communiquant avec Twitter a accès aux données exactes

de localisation d'émission du tweet. Cependant, bien souvent, l'utilisateur ne partage pas sa position de façon automatisée mais l'indique manuellement. Dans ce cas, cette position est enregistrée dans *<author_location>*, sachant que cette déclaration, si elle est valide, n'est pas pour autant sûre puisque l'utilisateur peut déclarer tout et n'importe quoi.

## 4.4 Exemple de création de corpus

La création de corpus par notre interface permet de considérer les trois dimensions constituant pour nous le cœur d'un événement à savoir les actions recherchées (définies par des mots-clés, la fenêtre temporelle et la zone géographique d'intérêt. Par exemple, nous avons besoin de tweets sur *Nice* et *La Défense*. Une première approche est de lancer une requête avec une *Bounding_Box*[17] car l'API Twitter permet de lancer une requête en utilisant la longitude et la latitude de quatre points entourant une zone géographique donnée. Malheureusement, le résultat est bien trop décevant puisque la plupart des utilisateurs ont tendance à ne pas autoriser l'affichage de leurs coordonnées géographiques en twittant. En outre, si on ajoute la contrainte de Twitter qui ne permet de récupérer que 1% des tweets d'une requête donnée, nous obtenons un nombre de tweets insuffisant. D'après nos premiers tests, une première constatation a acté le fait que peu de tweets correspondent aux événements à étudier.

Notre deuxième approche consiste à mettre les paramètres de lieu à l'intérieur de la requête en combinant des identifiants de lieux comme *"@villedenice"* avec des articles de lieux comme "à proximité". Cette approche nous a permis de trouver beaucoup plus de tweets en accord avec nos événements malgré l'absence dans la plupart des tweets de toute indication de lieu dans les métadonnées, et ceci en moins de temps par rapport à une requête par *Bounding_Box* (131 tweets en 5 minutes, contre 1908 en 24h pour une requête par *Bounding_Box*).

Lors de la deuxième phase de notre constitution de corpus, nous avons utilisé des requêtes combinant des noms de lieu avec *La Défense* et *Nice* (par exemple, *nice, villedenice*, nom

17. voir `https://www.flickr.com/places/info/7153328`

de rivière dans la région, etc.) avec des mots-clés entourant un événement donné, comme le lecteur pourra le constater dans le tableau 4.1. Ici aussi, la contrainte de l'API Twitter nous a empêché d'avoir un nombre suffisant de tweets pour pouvoir déterminer tous les éléments à extraire (voir tableau 4.1).

Le nombre de tweets trouvés par nos requêtes étant insuffisant, nous avons décidé de ne plus spécifier le lieu mais plutôt de récupérer des tweets en spécifiant des mots-clés pour chaque événement. En effet, en français, il y a souvent un même *pattern* pour exprimer un événement donné, c'est-à-dire par l'emploi d'un ensemble de mots-clés et de verbes déclencheurs. Nous pouvons donc faire notre étude sur des tweets écrits en français sur un événement donné (*4.4.1*).

Par ce biais, nous pouvons sélectionner des tweets bien formés et bien écrits sur différents événements. Cela va nous permettre de passer à notre prochaine étape de recherche qui est la restructuration de l'ontologie de GEOLSemantics.

Les événements à étudier sont pour l'instant fixés selon les termes : *accident, bagarre, chute d'arbre, colis suspect, organisation de fêtes, glissement de terrain, incendie, inondation et dépôt d'ordures*. Bien évidemment, la représentation est assez générique pour pouvoir intégrer bien d'autres concepts, enrichissant ainsi la taxonomie de base.

### 4.4.1 Raffinement des requêtes de recherche

Afin de récupérer un corpus sur Twitter, une interrogation avec un ou plusieurs mots-clés est possible, quel que soit l'outil utilisé (Twint ou API Twitter). Des opérateurs servant de filtre sont utilisables pour raffiner la requête :

**OR** : opérateur "OU" indiquant une condition alternative ; exemple : la requête contenant le filtre *Nice OR Paris* renvoie les tweets qui ont "Nice" ou "Paris" dans le texte,

**AND** : opérateur "ET" indiquant que toutes les conditions de recherche doivent être satisfaites ; exemple : *Nice AND Paris* renvoie les tweets qui ont obligatoirement "Nice" et "Paris" dans le texte dans n'importe quel ordre,

| Corpus récupéré | | | | | Corpus sélectionné |
|---|---|---|---|---|---|
| Geoloc requête | Type requête | | Requête | Nombre tweets | Nombre tweets |
| Nice | Lieu | | (@villedenice OR nice )AND( proche OR vers OR a proximite de ) | 131 | 109 |
| | | | BoundingBox( Provence-Alpes-Cote dAzur ) | 1908 | |
| | Événement | Accident | ( nice OR @VilledeNice )AND( accident ) | 6 | 22 |
| | | Inondation | ( bragues OR cagne OR Loup ) AND (inondation OR crue centennale OR crues torrentielles ) | 24 | |
| | | | ( Malvan OR paillon OR promenade Riou de l'Argentiere )AND( ruissellement pluvial ) | 12 | |
| | | | ( Siagne OR vallons de Gambetta OR vallons de Magnan OR vallons de Rivoli OR var ) | 179 | |
| La Défense | Lieu | | ( ladefense OR courbevoie OR puteaux OR courbevoisiens OR les4Temps OR cnit ) | 478 | 109 |
| | | | ( ladefense ) AND ( tour OR building OR place OR vers ) | 13 | |
| | | | ( ladefense ) AND ( rer OR bus ) | 9 | |
| | Événement | Inondation | ( accès OR annulé OR attention OR détruit OR fermé )AND( rafales OR inondation )AND( puteaux OR berges OR paris OR Seine ) | 1069 | 93 |
| ND | Événement | Accident | ( accident ) | 807 | 153 |
| | | | ( embouteillage )OR( bouchon -boire ) | 1330 | |
| | | Arbre | ( inondation OR tempête ) AND ( arbre arraché OR arbre tombé OR déborde -énergie ) | 11046 | 2 |
| | | Colis | ( colis ) AND ( abandonné OR suspect ) | 447 | 55 |
| | | Incendie | ( feu ) OR ( fumée ) | 70 | 81 |
| | | | ( incendie OR feu-rouge OR brul... OR fumer Or flammes ) | 4315 | |
| | | Incident | ( incident ) | 262 | 69 |
| | | Inondation | ( déborde ) | 1383 | 69 |
| | | | ( inondation ) | 1061 | |
| | | | ( intempérie ) | 36 | |
| | | Malaise | (( avoir OR eu ) AND ( un malaise)) OR ( dans les pommes ) OR ( dans les vapes ) OR ( perd... connaissance ) OR ( évanoui) | 22326 | 0 |
| | | Ordure | ( ordure déposée ) | 193 | 4 |
| | | | ( ordure OR détritus ) | 194 | |
| | | Organisation fêtes | ( rave party ) | 129 | 18 |
| | | Route | ( route coupée ) | 60 | 0 |
| | | | ( accès ) AND ( coupé OR fermé ) | 247 | |

TABLEAU 4.1 – Corpus : requête de récupération des tweets pour chaque événement

( ) : opérateur de priorisation des critères de recherche ; exemple : *Nice OR (Paris AND ville)* renvoie les tweets contenant Nice ou le couple de termes (Paris, ville) dans le texte du tweet, alors que la même requête sans parenthèse renvoie les tweets avec

— soit seulement "Nice" dans les tweets ;

— soit des tweets avec obligatoirement "Paris" et "ville" dans les tweets,

“ ” : opérateur de verbatim pour avoir l'expression exacte dans un ordre précis ; exemple : *“Ville de Paris”* renvoie les tweets avec ville de Paris dans cet ordre,

- : opérateur pour ignorer un mot ou une expression ; exemple : *accident -vasculaire* renvoie les tweets contenant le terme *accident* mais pas les tweets avec le groupe de mots *accident vasculaire*.

Il est néanmoins très difficile d'avoir des tweets sur un événement donné en utilisant des mots-clés concernant cet événement. En effet, un mot peut avoir plusieurs sens selon le contexte. En outre, l'utilisation des métaphores rend la tâche encore plus difficile (cf. figure 4.3).

Dans ce qui suit, nous allons essayer de trouver la requête la plus pertinente et la plus robuste renvoyant le moins de bruit possible pour chaque action prévue, et ce, afin de créer un corpus pour chaque action dans le territoire de la ville de Nice.

Nous avons commencé par exécuter des requêtes en utilisant des mots-clés sur les événements sélectionnés, puis nous avons annoté manuellement le résultat et enfin, nous avons sélectionné les tweets qui parlent vraiment d'événements. Le tableau 4.2 représente le résultat de cette étape avec une case représentant la requête par mots-clés de chaque événement, des exemples de tweets qui ont été retenus et de tweets ignorés. Enfin, nous indiquons le nombre de tweets retenus par rapport au nombre de tweets collectés.

Les mots-clés peuvent ne pas être pertinents puisque souvent le contexte est indispensable pour trouver le bon tweet. Par exemple, le mot-clé *accident* seul peut renvoyer des tweets concernant *accident vasculaire*, alors que l'expression *accident de voiture* ne va renvoyer que des tweets d'accident de voiture.

Étant donné que le nombre de tweets pertinents est minime comme le stipule la case "nombre de tweets retenus" dans le tableau 4.2 (seulement 10% des tweets retenus), nous




FIGURE 4.3 – Exemple d'un tweet d'un accident (métaphorique)

avons décidé de transformer chaque tweet retenu en requête (*i.e.* requête par expressions-clés) en ne gardant que la partie qui fait apparaître l'événement. Par exemple, *"Accident mortel à Saint-Émilion : un motard percute un engin agricole"* sera transformé en deux requêtes

— *1) "Accident mortel à"*

— *2) "un motard percute".*

Une fois cette transformation faite, on récupère pour chaque requête au moins 100 tweets. Le tableau 4.3 présente les nouveaux résultats.

Le tableau 4.3 montre que 66% des tweets sont retenus. Puisque certains éléments du contexte (ou mots-outils) peuvent générer du bruit, nous allons former une requête par mot-clé mais en deux temps ; une partie avec les mots qui sont en relation directe avec l'événement et une partie qui ajoute une information supplémentaire. Par exemple, si *accident* est la partie primaire et *voiture* la secondaire, alors la requête sera *(accident)*

*AND (voiture).*

Pour réaliser cela, nous avons récupéré pour chaque corpus d'événements retenus, les mots-clés qui ont le plus grand nombre d'occurrences (sauf mots-outils). Nous les avons classés en "mots-clés directs" et "mots-clés supplémentaires" afin de créer une requête plus robuste pour chaque action. Les informations pour chaque événement sont indiquées dans le tableau 4.4.

Une fois les requêtes construites, nous avons commencé à récupérer les tweets sur le territoire de Nice en appliquant deux méthodes :

— par *bounding box* : nous avons utilisé les requêtes du tableau 4.4 en utilisant une *bounding box* sur la zone de la ville de Nice en utilisant l'API Twitter, il est à noter que tous les tweets résultant de cette méthode sont des tweets de personnes qui ont autorisé le partage de leurs coordonnées GPS

— par nom de lieu : nous avons récupéré tous les lieux de Nice présents dans la base de données *Openstreetmap* (84905 lieux) puis nous avons lancé la requête en les combinant avec le résultat du tableau 4.4. Par exemple, une requête peut être exprimée comme suit : *" Zone d'activité 40 chemin de la Parette " AND (dechet OR ordure OR poubelle) AND (sauvage OR dépôt OR Environment)*. Nous avons choisi de récupérer 100 tweets au maximum pour chaque requête pour notre première étude.

## 4.5 Corpus d'expérimentation

Nous avons vu dans la section 4.4.1 comment affiner une requête pour avoir des tweets sur des événements avec le moins de bruit possible ; c'est-à-dire des tweets avec des mots-clés en relation avec un événement donné mais qui ne sont pas des tweets sur cet événement. Nous avons vu aussi comment mettre en pratique une requête sur un territoire donné en lançant la récupération des tweets par *bounding box*, et par nom de lieu sur la zone de Nice. Nous allons analyser dans ce qui suit les corpus que nous avons obtenus grâce à ces méthodes.

### 4.5.1 Obtention de tweets par nom de lieu

Nous avons lancé la récupération des tweets sur la période d'existence de Twitter (à partir de 2006), soit 84 906 requêtes pour chaque événement, chaque requête étant composée d'un nom de lieu dans Nice et de la requête affinée de chaque événement (cf. sous-section 4.4.1).

**Exemple 4.5.1.** Soit la requête *" Boulevard du Docteur Bourgarel " AND (bagage OR bombe OR colis OR valise OR sac) AND (Alerte OR abandonné OR suspect OR Périmètre sécurité OR déminage)*

Le nombre de tweets obtenus avec ou sans doublon selon l'événement recherché sont présentés dans le tableau 4.5.

Sur toute la période analysée, beaucoup d'événements sont collectés mais l'organisation de fêtes est l'événement le plus fréquemment rencontré avec 28 576 tweets.

Nous avons ensuite annoté 20% des tweets pour observer le pourcentage de tweets qui ne parlent pas d'événements 4.8.

après l'annotation manuelle on a trouvé que 100% des tweets récupérés sur l'événement « glissements de terrain » était corrects, tandis que pour « ordure sauvage » nous n'avons pu récupérer que 68% de tweets corrects. 4.8

Nous allons essayer de voir dans la sous-section 4.5.2 si en appliquant les mêmes requêtes sur une *bounding box* autour de la ville de Nice, nous pourrons collecter ces événements avec une précision plus grande sur le lieu.

### 4.5.2 Récupération des tweets par bounding box

Nous avons lancé la récupération des tweets sur la période entre le 21-08-2020 et le 14-09-2020 sur une *bounding box* dessinée sur la zone de Nice. Nous avons obtenu 211 863 tweets. Nous avons ensuite appliqué les requêtes affinées (voir section 4.4.1) de chaque événement pour obtenir les tableaux 4.9, 4.10, 4.11, 4.12.

Nous pouvons voir dans le tableau 4.9 que le nombre de tweets relevant d'un événement

est minime par rapport au flux Twitter. L'événement le plus commun est organisation de fête avec 164 tweets sur 211.863, il représente 0.08% des tweets.

Dans le tableau 4.10 nous pouvons remarquer que dans les tweets relatant un événement, la moyenne des caractères est souvent de plus de 100 caractères, tandis qu'elle est en dessous de 80 pour les tweets les plus généraux.

Nous pouvons remarquer aussi que l'utilisation des hashtags et des mentions est plus fréquente en cas d'événement que sur les discussions générales. Seule exception pour organisation de fêtes (moyenne de # : 0.27 par tweet, moyenne de mention : 0.50 par tweet) qui a tendance à ne pas vouloir être trop visible pour éviter d'être annulée.

Dans le tableau 4.11, nous pouvons remarquer que les hashtags utilisés sont cohérents avec le contenu : *#feu* pour incendie et *#pluiesdiluviennes* pour inondation, tout comme les mentions : *@SDIS83 (SDIS - POMPIERS)* pour incendie et *@Proprete0 (Propreté & collecte à Nice)* pour les dépôts d'ordures.

Par contre, nous pouvons remarquer que, malgré la contrainte de la *bounding box*, l'actualité se déplace ; comme on le voit avec le hashtag *#PSGOM* qui est utilisé pour inciter les gens à sortir célébrer la victoire de l'OM contre le PSG (match joué le 13 septembre 2020).

Dans tous les tweets relevés, aucun n'a mentionné la police pour lancer l'alerte sur un des événements.

Enfin, nous pouvons remarquer dans le tableau 4.12 que les requêtes affinées entraînent toujours beaucoup de bruit. Par exemple, aucun des 26 tweets renvoyés par la requête on cherchait les chutes d'arbre ne représentent réellement l'événement « chute d'arbre ». En effet, dans ce cas précis, c'est l'utilisation de nombreuses métaphores qui a engendré cette mauvaise interprétation.

## 4.6 Premières constatations

Nos premières constatations lors de la constitution du corpus [18] concernent principalement les convergences ou divergences des tweets du corpus selon les méthodes de collectes mais aussi l'orthographe utilisé dans les tweets.

### 4.6.1 Analyse des tweets collectés

Nous pouvons remarquer dans la colonne *nb doublons* du tableau 4.5 qu'un nombre de doublons considérable apparaissent dans les tweets collectés. Le plus souvent, il s'agit de réponses génériques courtes comme *lol* ou des émoticônes. Cependant, nous avons aussi observé un phénomène inattendu concernant des personnes qui ne retweetent pas mais vont copier un tweet et le tweeter tel quel comme si elles en étaient l'auteur. Cette technique se trouve souvent sur des comptes gérés automatiquement par un algorithme [Cramer and Büttner, 2011].

Dans le tableau 4.6, nous pouvons remarquer que le nombre moyen de caractères utilisés pour les tweets d'événement se situe entre 166 et 228 caractères ; cette moyenne est à prendre avec précaution compte-tenu que sur la période 2006 à 2017, les tweets ne pouvaient pas excéder 144 caractères.

En effet, 62 792 tweets récupérés dataient d'avant novembre 2017 (date du passage à 280 caractères) et avaient une moyenne de 120 caractères par tweet. Les 21 714 864 tweets récupérés après novembre 2017 avaient une moyenne de 215 caractères par tweet. Cette moyenne de 215 nous montre que certains événements sont exprimés avec plus de caractères que d'autres. Cela est aussi représentatif du nombre d'abonnées qui est passé de 173 000 en 2010 à 1 318 000 en 2018 comme indiqué dans la figure 4.4 [19].

Les événements « colis suspect », « inondation » et « incendie » sont toujours accompagnés d'au moins un *hashtag* en moyenne. Les mentions (qui débutent par @) sont moins présentes allant d'une moyenne de 0,17 par tweet dans le glissement de terrain à 0,33 pour

18. téléchargeable à linc.iut.univ-paris8.fr/eventcorpus
19. `https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users`

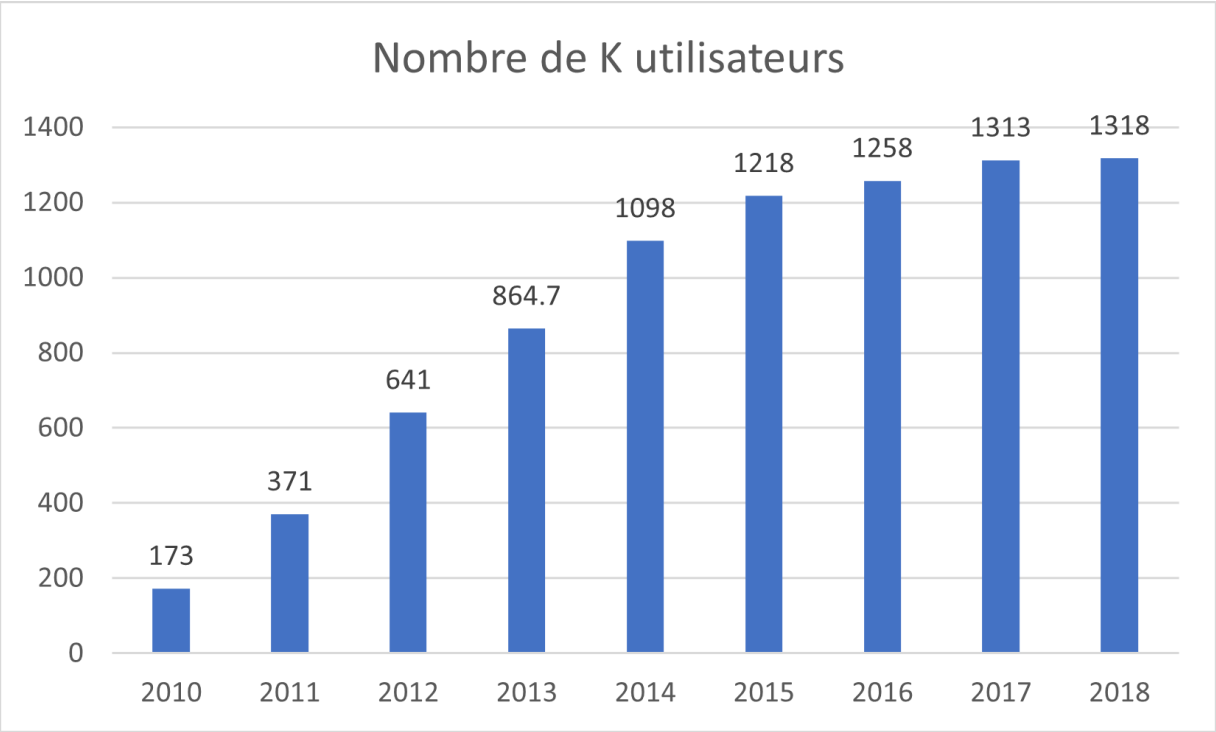



FIGURE 4.4 – Nombre d'utilisateurs de Twitter actifs dans le monde de 2010 à 2018

l'organisation de fêtes. Nous détaillerons notre analyse dans la sous-section 4.6.1.

En ce qui concerne les *hashtags* utilisés, *#ColisSuspect* est le plus fréquent pour l'événement « colis suspect », *#police* pour « bagarre » , *#pompier* et *#inondation* pour « incendie » et « inondation » respectivement, comme révélé dans le tableau 4.7.

Dans 202 mentions sur 7 109 (cf. tableau 4.7), les twittos mettent les services à prévenir dans leurs tweets. Cela appuie ce que nous souhaitons effectuer dans le projet puisque cela reviendrait, pour le reste des tweets sans mention, à ajouter une mention des services concernés aux tweets.

L'événement « chute d'arbre » était plus orienté vers la ville de Paris avec le *hashtag* et la mention du *@rerA*. Beaucoup de noms de lieux de Nice existent aussi dans d'autres villes, ce qui explique la présence d'événements indiquant d'autres lieux dans les tweets récupérés avec des noms de lieux à Nice.

Nous avons ensuite cherché à savoir si les gens lancent une alerte par eux-mêmes en mentionnant le compte des autorités, et, dans l'affirmative, quelle est cette autorité. Le tableau 4.13 récapitule cela. Nous avons stocké des exemples pour chaque événement ; ce

qui apparaît dans le tableau 4.14.

Dans le tableau 4.13, nous pouvons remarquer que les twittos ont tendance à signaler des événements à la ville en priorité, puis à la police. Les pompiers sont sollicités principalement pour les incendies.

Bien que les événements récupérés étaient corrects dans la plupart des cas (une moyenne de 91,94%), les lieux étaient moins précis, notamment à cause de noms de lieux qui peuvent exister dans plusieurs villes de France mais aussi de noms de lieux trop généraux comme « la piscine » ou « station service » par exemple (cf. tableau 4.8.

Nous allons donc supposer que tous les tweets qui vont nous servir lors de nos expérimentations sont correctement annoté malgré la marge d'erreur de presque 10%. Cette marge sera corrigée pour le corpus d'apprentissage au fur et à mesure des développements. Quant au corpus de test, il sera constitué du corpus qui a été annoté manuellement.

### 4.6.2 Constatations sur l'orthographe

Les tweets générés pour le projet *"SafeCity"* nous ont permis de constater une syntaxe simple (sujet, verbe, complément) et une orthographe assez correcte (peu d'écriture phonétique) mais des confusions systématiques entre participe passé et infinitif, entre terminaison de verbes "s-t-ent", "ons-ont", entre "ou" et "où", "a" et "à", etc. L'amélioration des traitements linguistiques devra prendre en considération les aspects orthographiques ci-dessus afin de trouver les bonnes interprétations syntaxiques et sémantiques des mots.

Pour ce qui concernent les mots écrits phonétiquement, ils sont inconnus du dictionnaire français actuel sur lequel se fondent les analyseurs de GEOLSemantics. Une analyse des interprétations phonétiques les plus courantes va permettre de transformer les mots trouvés en une forme normalisée. Elle sera aussi utilisée pour les mots du dictionnaire permettant de suggérer des interprétations. De plus, les expressions courantes telles que des expressions figées (par exemple, *"Le mariage pour tous"*), des proverbes (par ex, *"Il ne faut pas vendre la peau de l'ours avant de l'avoir tué."*), etc., seront ajoutées dans le dictionnaire.

## 4.7 Conclusion

Nous avons vu dans ce chapitre que la constitution d'un corpus de tweets est une tâche aussi utile que délicate, que ce soit pour la collecte des tweets, leur annotation ou le traitement et la publication.

Bien que l'affinage de la requête nous ait permis de récupérer des tweets sur un domaine donné avec une marge d'erreur acceptable, la récupération des tweets sur une zone déterminée reste une tâche très difficile. Par exemple, en utilisant une *bounding box* sur une zone déterminée, on récupère quand même des tweets sur des événements en dehors de cette zone puisque les gens peuvent parler d'événements extérieurs. Même si le lieu le plus pertinent reste celui qui est dans le texte du tweet, il arrive parfois qu'un terme peut définir plusieurs lieux dans plusieurs régions d'où la nécessité d'une désambiguïsation géographique.

Nous allons dans le chapitre 7.4 tester notre solution sur les corpus constitués afin de vérifier le bien fondé de notre processus de traitements linguistiques de tweets pour obtenir une meilleure précision sur le couple (action, lieu) qui sont les deux premiers éléments de notre triplet définissant un événement. L'identification complète de l'événement n'en sera que plus facilité.

| Événement | Requête | Exemple à retenir | Exemple à écarter | Nombre de tweets retenus |
|---|---|---|---|---|
| accident | (“accident” OR “collision” OR “percute” )<br>-“du travail” -“par accident”<br>-“risque d’accidents”<br>-(domestiques OR vasculaires OR cérébral) | 08h31 : En direction de Narbonne<br>après Echangeur 15 La Roseraie<br>#Accident. La circulation est perturbée. | Je n’avais pas percuté jusqu’à là,<br>trouvant tout naturelle qu’à la télé on ne<br>voit que des blancs dans les pubs<br>comme si les noirs n’existaient pas dans notre société. | 73/992 |
| bagarre | (“bagarre” OR “altercation”<br>OR “coup de feu” OR “bracage“ OR “explosion”)<br>-(joie OR colère OR chômage OR délinquance) | Une bagarre sanglante éclate dans<br>un centre commercial d’orly | Par contre avec les verts sa va être<br>un retour en force des gilets jaunes et<br>une explosion des impôts | 25/1000 |
| chute<br>d’arbre | “chute d’un arbre” OR “arbre tombé”<br>OR “chute d’arbre” OR “arbre tombé”<br>“chute d’une branche” OR “branche tombée”<br>OR “chute d’une branche” | Trois enfants blessés par la<br>chute d’une branche de banian à St_Pierre | Campagne de sensibilisation et de<br>conscientisation LES ARBRES S’EXPRIMENT.<br>Un arbre tombe, plantes en un autre Proverbe birman | 300/1863 |
| Colis<br>Suspect | “#ColisSuspect” OR “#valise_suspecte”<br>OR ((paquet OR sac OR valise OR paquet)<br>AND (supect OR délaissé OR oublié)) | Infotrafic Fin du colis suspect<br>à Duchesse Anne | Job il est trop insolent je lui dis<br>que je suis avec son Reuf il me dis<br>« c’est pas mon Reuf c’est un colis suspect » | 300/3008 |
| organisation<br>de fêtes | “teuf” OR “rave party” OR “une rave”<br>OR festoche OR festoch OR<br>(“fête” -“des pères” -“de mères”) | Et mes voisins du dessus ont<br>visiblement organisé une rave party,<br>mon plafond vibre dangereusement et y’a<br>quelques bouts de plâtres qui en tombe | Bah ils vont nous faire une "Balkany" :<br>ça va trainer encore quelque années,<br>il finira par aller en taule pendant quelques<br>semaines avant d’être libéré parce que la<br>déco ne lui conviendra pas (et pour aller guincher à<br>la fête de la musique de Sablé sur Sarthe). | 57/2352 |
| glissements<br>de terrain | “glissement de terrain” | Deux disparus après un glissement<br>de terrain dans le comté de<br>Danba, dans le sud_ouest de la #Chine | Nouvelle tendance lifestyle : le glissement de<br>terrain challenge. pic.twitter.com/t0zvGBCcZG | 69/1010 |
| incendie | incendie -“détection” -“centre” -“service” -“anti”<br>-“sécurité incendie” -“risque d’incendie” -“alarme” | y’a le feu à côté de chez moi<br>pic.twitter.com/yJtqtb2gQe | Y’a le feu dans les comptes de Jacques Henri | 76/2000 |
| inondation | (inondation OR inondé OR déborde) -<br>“déborde de” -“inondé de” | #Événement #Toronto #AUT401 Ouest<br>bretelle d’accès à Markham Rd Sud,<br>toutes les voies sont bloquées en raison<br>d’inondation. #ONRtes | a 253k fav j’achète un poisson et j’inonde<br>ma chambre pour faire comme un aquarium | 6/1006 |
| dépôt<br>d’ordures | “dépôt d’ordure” OR “dépôt de gravats” | #Cugnaux# s’arrange, face à la rue<br>Jules Valles, un dépôt d’ordures depuis<br>3 semaines.... qu’elle image de la ville accueillante ?.... | t’es tellement moche que quand tes parents te<br>déposent le matin au lycée<br>ils sont facturés pour dépôt d’ordure | 23/1005 |
| Total | | | | 929/9389 |

TABLEAU 4.2 – Récupération des tweets en utilisant des mots-clés

| Événement | Requête | Exemple à retenir | Exemple à écarter | Nombre de tweets retenus |
|---|---|---|---|---|
| accident | Requête composée d'un tweet événement | A 1 Lausanne >Yverdon entre Echangeur d'Essert Pittet et Echangeur d'Yverdon ralentissements, accident, voie de gauche bloquée | Un corps a été retrouvé flottant près du Four à Chaux à Labattoir par des pêcheurs. la gendarmerie et les pompiers se rendent sur place. Aucune confirmation pour l'instant que c'est le corps du comédien Kham's disparu hier. | 2079/3021 |
| bagarre | Requête composée d'un tweet événement | Saint Benoît : Une bagarre et des blessures au sabre sur fond d'alcool | FLASH GAZA : Un palestinien a été grièvement blessé à la tête par une balle d'un sniper de l'armée israélienne, à l'est de la ville de Khan Younès, dans le sud de la Bande de Gaza. | 107/364 |
| chute d'arbre | Requête composée d'un tweet événement | Un arbre est tombé sur les voies entre Bréauté et Fécamp. | e duo chic et choc lors de la réunion les Panafricaines. Pour reprendre un proverbe africain, Quand un arbre tombe, on l'entend ; quand la forêt pousse, pas un bruit. | 2553/2720 |
| colis suspect | Requête composée d'un tweet événement | Lyon : la gare de la Part Dieu évacuée en raison de deux colis suspects via le parisien | Fin d'incident en gare de Bobigny Pablo Picasso. le trafic reprend progressivement sur la ligne. ligne5 RATP | 1528/3919 |
| organisation de fêtes | Requête composée d'un tweet événement | Partir à la campagne pour être au calme ce week end, découvrir qu'une rave party aura lieu ce soir dans les bois au dessus de la maison.. Des gendarmes partout | 1 coup de gueule.. j 'ai 73 ans, seule à la campagne pour confinem. que je respecte 3 sorties en 1 mois, mais pensez vous que les vieux dans les Résid. Séniors soient responsables ils ne sont pas aller au foot, dans une rave ou faire du jogging.. alors NON à + de confinem. | 179/319 |
| glissements de terrain | Requête composée d'un tweet événement | SNCF : Un TER sur 2 en circulation entre Martigues et Miramas à cause d'un glissement de terrain à Port de Bouc | L'Indonesie de nouveau frappé par un tsunami : au moins 168 morts. Le bilan de ce tsunami, sans doute causé par un glissement de terrain sous marin près du détroit de la Sonde | 727/865 |
| incendie | Requête composée d'un tweet événement | Yonne Un impressionnant incendie est toujours en cours à l'ancienne fromagerie Paul Renard, à Flogny la Chapelle. L'épaisse fumée noire qui s'échappe des bâtiments industriels est visible depuis Joigny, Carisey, Sacy Périmètre de sécurité en place | Donc je suis sur le feu d'artifice de France2 qu'ils ont la (bonne) idée de diffuser sans bruit ambiant (un peu nul). Mais y'a le feu de Disney qui pete à côté de chez moi et que j'entends bien | 1113/1349 |
| inondation | Requête composée d'un tweet événement | la rivière Chaudière déborde et inonde 230 maisons ainsi que des commerces à Beauceville au Québec | INONDÉ PAR LA LUMIÈRE NATURELLE Fenêtres plein pieds, balcons surdimensionnés | 292/330 |
| dépôt d'ordures | Requête composée d'un tweet événement | Villedebegles Bx Metro signalement : dépôt d'ordures sauvages avec seringues sur passerelle SNCF impasse des camps Begles Côté intersection rue Lyautey et Fayolle secteur très fréquenté par les enfants. Pouvez vous intervenir ? | zeghostw Sur l'echangeur pres de l'hotel olympe, j'etais en hauteur jme suis cogné contre la barrière de protection | 18/28 |
| Total | | | | 8596/12915 |

TABLEAU 4.3 – Récupération des tweets par expressions-clés

| Événement | mots-clés directs nb. occurrences | mots-clés complémentaires nb. occurrences | Requêtes |
|---|---|---|---|
| accident<br>8932 mots<br>34659 occurrences | accident 1962<br>collision 61<br>désincarcération 53 | camion 59<br>voie 148<br>voiture 363<br>victime 73<br>véhicule 137<br>moto 143<br>direction 249 | (accident OR collision OR désincarcération)<br>AND<br>(camion OR voie OR voiture OR victime OR véhicule OR moto OR direction) |
| bagarre<br>873 mots<br>1541 occurrences | bagarre 89<br>dispute 2<br>rixe 10 | blessé 22<br>mort 17<br>police 12<br>alcool 80<br>violent 9 | (bagarre OR dispute OR rixe)<br>AND<br>(blessé OR mort OR police OR alcool OR violent) |
| chute<br>10479 mots<br>40517 occurrences | arbre 2292<br>branche 666 | chute 1361<br>tombé 841<br>blessé 924<br>trafic 1333<br>tué 213<br>circulation 213<br>mort 84<br>Tempête 112 | (arbre OR branche)<br>AND<br>(chute OR tombé OR blessé OR trafic OR tué OR circulation OR mort OR Tempête) |
| Colis Suspect<br>5603 mots<br>22376 occurrences | bagage 54<br>bombe 59<br>colis 1291<br>valise 85<br>sac 7 | Alerte 149<br>abandonné 38<br>suspect 1297<br>Périmètre 137<br>sécurité 152<br>déminage 165 | (bagage OR bombe OR colis OR valise OR sac)<br>AND<br>(Alerte OR abandonné OR suspect OR Périmètre sécurité OR déminage) |
| fête<br>1169 mots<br>2181 occurrences | CONCERT 10<br>fête 21<br>party 15<br>rave 14<br>teuf 39 | sauvage 3<br>Détails 14<br>voisins 18 | (CONCERT OR fête OR party OR rave OR teuf)<br>AND<br>(sauvage OR fêter OR Détails OR voisins) |
| glissement<br>4057 mots<br>10286 occurrences | affaissement 21<br>glissement 715 | terrain 803<br>terre 21<br>évacué 22<br>emporté 35 | (affaissement OR glissement)<br>AND<br>(terrain OR terre OR évacué OR emporté) |
| incendie<br>5839 mots<br>14633 occurences | feu 984<br>flammes 17<br>incendie 147<br>fumée 116 | pompier 174<br>victime 21<br>secours 14 | (feu OR flammes OR incendie OR fumée)<br>AND<br>(pompier OR victime OR secours) |
| inondation<br>2291 mots<br>4624 occurences | inondation 155<br>inondé 130 | rivière 86<br>déborde 14<br>ALERTE 96<br>pluie 50<br>orage 22 | (inondation OR inondé)<br>AND<br>(rivière OR déborde OR ALERTE OR pluie OR orage) |
| ordures<br>294 mots et 351 occurrences | déchet 1<br>ordure 6<br>poubelle 2 | sauvage 4<br>dépôt 18<br>Environnement 1 | (déchet OR ordure OR poubelle)<br>AND<br>(sauvage OR dépôt OR Environnement) |

TABLEAU 4.4 – Création des requêtes avec contexte

| Nom de l'événement | nb tweets | sans doublons | nb doublons |
|---|---|---|---|
| incendie | 23 035 | 22 831 | 204 |
| inondation | 8 420 | 8 274 | 146 |
| organisation de fêtes | 28 576 | 28 434 | 142 |
| dépôt d'ordures | 855 | 844 | 11 |
| accident | 64 969 | 64 525 | 444 |
| bagarre | 9 662 | 9 588 | 74 |
| chute d'arbre | 21 580 | 20 772 | 808 |
| colis suspect | 3 774 | 3 691 | 83 |
| glissements de terrain | 877 | 872 | 5 |
| Total | 161 748 | 159 831 | 1 917 |

TABLEAU 4.5 – Nombre de tweets obtenus par type d'événements pour *Nice*.

| Nom de l'événement | nb tweets | nb caractères | moyenne caractères par tweet | nb # | nb @ | nb de # par tweet | nb de @ par tweet |
|---|---|---|---|---|---|---|---|
| incendie | 23 035 | 4 475 054 | 196 | 21 768 | 7 109 | 0.95 | 0.31 |
| inondation | 8 420 | 1 528 129 | 185 | 12 509 | 2 126 | 1.51 | 0.26 |
| organisation de fêtes | 28 576 | 5 177 898 | 182 | 19 611 | 9 363 | 0.69 | 0.33 |
| dépôt d'ordures | 855 | 192 229 | 228 | 528 | 242 | 0.63 | 0.29 |
| accident | 64 969 | 11 496 906 | 178 | 55 400 | 12 886 | 0.86 | 0.20 |
| bagarre | 9 662 | 1 799 407 | 188 | 5 094 | 1 719 | 0.53 | 0.18 |
| chute d'arbre | 21 580 | 3 696 270 | 178 | 10 977 | 4 378 | 0.53 | 0.21 |
| colis suspect | 3 774 | 614 110 | 166 | 4 052 | 765 | 1.10 | 0.21 |
| glissements de terrain | 877 | 163 795 | 188 | 426 | 146 | 0.49 | 0.17 |
| Total | 161 748 | 29 143 798 | 188 | 130 365 | 38 734 | 0.81 | 0.24 |

TABLEAU 4.6 – Moyenne des caractères par tweet et par type d'événements pour Nice (bounding box)

| Nom de l'événement | nb tweets | mention la plus fréquente | nb mention de la police | le # le plus fréquent |
|---|---|---|---|---|
| incendie | 23 035 | @SDIS34 (202 fois) | 157 | #pompier (1 026 fois) |
| inondation | 8 420 | @meteofrance (95 fois) | 7 | #inondation (646 fois) |
| organisation de fêtes | 28 576 | @YouTube (92 fois) | 16 | #music (502 fois) |
| dépôt d'ordures | 855 | @Paris (4 fois) | 4 | #environment (32 fois) |
| accident | 64 969 | @Radio1077 (2303 fois) | 147 | #Accident (2177 fois) |
| bagarre | 9 662 | @f_desouche (47 fois) | 29 | #police (85 fois) |
| chute d'arbre | 21 580 | @RER_A (44 fois) | 12 | #RERA (131 fois) |
| colis suspect | 3 774 | @SNCF (20 fois) | 53 | #ColisSuspect (142 fois) |
| glissements de terrain | 877 | @SNCF (4 fois) | 1 | #TGV (12 fois) |
| Total | 161 748 | - | 426 | - |

TABLEAU 4.7 – Utilisation des mentions et de # par type d'événements pour Nice (bounding box)

| Nom de l'événement | nb tweets | nb de tweets annotés | % de tweets annotés | nb de vrais événements | % événements |
|---|---|---|---|---|---|
| incendie | 23 035 | 2 207 | 9,67 | 1 848 | 83,73 |
| inondation | 8 420 | 697 | 8,42 | 667 | 95,70 |
| organisation de fêtes | 28576 | 1597 | 5,62 | 1 554 | 97,31 |
| dépôt d'ordures | 855 | 60 | 7,11 | 41 | 68,33 |
| accident | 64 969 | 5 215 | 8,08 | 4 879 | 93,56 |
| bagarre | 9 662 | 585 | 6,10 | 569 | 97,26 |
| chute d'arbre | 21 580 | 3 480 | 16,75 | 3 196 | 91,84 |
| colis suspect | 3 774 | 1 637 | 44,35 | 1 633 | 99,76 |
| glissements de terrain | 877 | 734 | 84,17 | 734 | 100,00 |
| Total | 161748 | 16212 | 21,14 | 15 121 | 91,94 |

TABLEAU 4.8 – Pourcentage de vrais événements dans les tweets annotés pour Nice

| Nom de l'événement | nb tweets | sans doublons | nb doublons |
|---|---|---|---|
| incendie | 28 | 28 | 0 |
| inondation | 2 | 2 | 0 |
| organisation de fêtes | 164 | 163 | 1 |
| dépôt d'ordures | 1 | 1 | 0 |
| accident | 4 | 4 | 0 |
| bagarre | 2 | 2 | 0 |
| chute d'arbre | 26 | 26 | 0 |
| colis suspect | 0 | 0 | 0 |
| glissements de terrain | 0 | 0 | 0 |
| reste | 211 636 | 207 318 | 4 318 |
| Total | 211 863 | 207 544 | 4 319 |

TABLEAU 4.9 – Nombre de tweets obtenus par type d'événements sur une *bouding box* de *Nice*

| Nom de l'événement | nb tweets | nb caractères | moyenne caractères | nb # | nb @ | nb de # par tweet | nb @ par tweet |
|---|---|---|---|---|---|---|---|
| incendie | 28 | 3 622 | 129 | 28 | 49 | 1,00 | 1.75 |
| inondation | 2 | 264 | 132 | 3 | 0 | 1,50 | 0.00 |
| organisation de fêtes | 164 | 17 673 | 108 | 44 | 82 | 0,27 | 0.50 |
| dépôt d'ordures | 1 | 140 | 140 | 0 | 1 | 0,00 | 1.00 |
| accident | 4 | 427 | 107 | 0 | 0 | 0,00 | 0.00 |
| bagarre | 2 | 259 | 130 | 3 | 0 | 1,50 | 0.00 |
| chute d'arbre | 26 | 3 117 | 120 | 16 | 17 | 0,62 | 0.65 |
| colis suspect | 0 | 0 | 0 | 0 | 0 | 0,00 | 0.00 |
| glissements de terrain | 0 | 0 | 0 | 0 | 0 | 0,00 | 0.00 |
| reste | 211636 | 14 896 200 | 72 | 47 780 | 193 718 | 0,23 | 0.93 |
| Total | 211863 | 14 921 702 | 94 | 47 874 | 193 867 | 0,51 | 0,48 |

TABLEAU 4.10 – Moyenne des caractères par tweet et par type d'événements pour Nice (requête)

| Nom de l'événement | nb tweets | mention la plus fréquente | nb mention de la police | le # le plus frequent |
|---|---|---|---|---|
| incendie | 28 | @SDIS83 (5 fois) | 0 | #feu (3fois) |
| inondation | 2 | - | 0 | #pluiesdiluviennes (1 fois) |
| organisation de fêtes | 164 | @marseille (3 fois) | 0 | #PSGOM (4 fois) |
| dépôt d'ordures | 1 | @Proprete0 (1 foid) | 0 | - |
| accident | 4 | - | - | - |
| bagarre | 2 | - | - | #rixe (1 fois) |
| chute d'arbre | 26 | @InfosBordeaux (1 fois) | 0 | #Bordeaux (1 fois) |
| colis suspect | 0 | - | - | - |
| glissements de terrain | 0 | - | - | - |
| reste | 211636 | @FabouTaz (714 fois) | 324 | #KohLanta (442 fois) |
| Total | 211863 | | 324 | |

TABLEAU 4.11 – Utilisation des mentions et de # par type d'événements pour Nice (requête)

| Nom de l'événement | nb tweets | nb des tweets annoté | % des tweets annoté | nb évenements | % événements |
|---|---|---|---|---|---|
| incendie | 23035 | 2207 | 9.67% | 1848 | 83.73% |
| inondation | 8420 | 697 | 8.42% | 667 | 95.70% |
| organisation de fêtes | 28576 | 1597 | 5.62% | 1554 | 97.31% |
| dépôt d'ordures | 855 | 60 | 7.11% | 41 | 68.33% |
| accident | 64969 | 5215 | 8.08% | 4879 | 93.56% |
| bagarre | 9662 | 585 | 6.10% | 569 | 97.26% |
| chute d'arbre | 21580 | 3480 | 16.75% | 3196 | 91.84% |
| colis suspect | 3774 | 1637 | 44.35% | 1633 | 99.76% |
| glissements de terrain | 877 | 734 | 84.17% | 734 | 100.00% |
| Total | 161748 | 16212 | 21.14% | 15121 | 91.94% |

TABLEAU 4.12 – Pourcentage des tweets événement/le nombre de tweets annotés

| | police | | pompier | | samu | | ville | |
|---|---|---|---|---|---|---|---|---|
| | nb tweets | num ex4.14 | nb tweets | num ex4.14 | nb tweets | num ex4.14 | nb tweets | num ex4.14 |
| incendie | 157 | 4 | 364 | 13 | 9 | 19 | 191 | 22 |
| inondation | 7 | 5 | 15 | 14 | 0 | * | 63 | 23 |
| organisation de fetes | 16 | 6 | 1 | 15 | 0 | * | 222 | 24 |
| dépôt d'ordures | 4 | 7 | 0 | * | 0 | * | 11 | 25 |
| accident | 147 | 1 | 73 | 10 | 7 | 17 | 222 | 20 |
| bagarre | 29 | 8 | 0 | * | 0 | * | 14 | 26 |
| chute d'arbres | 12 | 9 | 8 | 16 | 0 | * | 93 | 27 |
| colis suspect | 53 | 2 | 1 | 11 | 1 | 18 | 23 | 21 |
| glissements de terrain | 1 | 3 | 2 | 12 | 0 | * | 0 | * |
| Total | 426 | | 464 | | 17 | | 839 | |

TABLEAU 4.13 – Nombre de mention des autorités par événements

| num ex | exemple |
|---|---|
| 1 | @PoliceNtleCi @PSecours225 @Liamkinkey ATTENTION Suite pluie circulation difficile de ENP vers BINGERVILLE. Feux clignotants jusqu'à carrefour ABATTA. |
| 2 | 50 personnes, barres de fer, blessés. Police appelée depuis 15mn, et toujours personne. Une explication @prefpolice |
| 3 | intervention @sdis44 en cours gare routière place ricordeau #Nantes arbre sur caténaire -Automobilist... /@POLICENAT44 |
| 4 | Accident grave de la circulation, la victime évacuée par hélicoptère@prefpolice |
| 5 | Fin d'intervention d'@PoliceNat91 sur la gare RER de @VilledeVigneux. |
| 6 | En raison d'un glissement de terrain, une grue menace de s'effondré. Les habitants de 6 immeubles ont été évacués. Eviter le secteur, les équipes des @PompiersParis et @prefpolice sont sur place. |
| 7 | #intervention @PompiersParis @PoliceNationale suite #incendie rue de Longchamp présente pour #aider #sinistrés |
| 8 | Tous les soirs en bas de chez nous 75 rue de Belleville des alcooliques font la fête, braillent et nous empêchent de dormir. Tous les soir avec des voisins nous appelons la @prefpolice |
| 9 | @PoliceNat67Dépôt sauvage d'ordure au sein de notre cour intérieure (quartier Marne) par un couple. |
| 10 | Les @PompiersFR des @AlpesMaritimes du secteur de #Nice sont sur place, des renforts sont en cours. |
| 11 | Pour fêter la #victoire de l'@equipedefrance, rien ne justifie de tout #casser, #voler ou encore mettre en danger sa #vie ou celles des autres.Faisons la #Fête ensemble, respectons les personnes qui assurent notre sécurité (@Gendarmerie @PoliceNationale @PompiersFR |
| 12 | @PompiersParis#Intervention Dans l'après-midi, les pompiers interviennent au quai Branly (XVe) pour la chute d'un arbre sur un véhicule. |
| 13 | #PRUDENCE sur les routes. Vers 5h30, les @pompiers24 sont intervenus sur la D6089. |
| 14 | opération de déminage d'une bombe de 250 kg terminée ! Bravo à l'équipe de déminage en lien avec @Saintcyrlecole, @Versailles, @AggloVGP, @PoliceNat78, @pompiers78, @SNCF, @GRDF et @samu78_ |
| 15 | @GPompiers alerté pr l'effondrement d'une barraque à Yop MICAO. Glissement de terrain ayant emporté la barraque |
| 16 | Les @Pompiers_Paris interviennent actuellement Porte de Saint-Cloud pour une importante inondation. |
| 17 | opération de déminage d'une bombe de 250 kg terminée ! Bravo à l'équipe de déminage en lien avec @Saintcyrlecole, @Versailles, @AggloVGP, @PoliceNat78, @pompiers78, @SNCF, @GRDF et @samu78_ |
| 18 | De nombreux moyens de secours du @SDIS85 et du @SAMU85 sont engagés sur un incendie à proximité de la piscine de la @VilledeLucon |
| 19 | Collision entre une voiture et un poids lourd sur le boulevard Willy-Brandt, à #Niort. Pompiers du @SDIS79, @PoliceNat79 et @SAMU7912 encore sur place. |
| 20 | Bonne ambiance à la fête des voisins ce samedi soir à Stalingrad ! @VilledeBagneux |
| 21 | @ville_albi base de cantepau arbre tombé [photo] |
| 22 | #Accident sur l'A10 (Paris >Pro) à hauteur de la sortie @VilledeDourdan. |
| 23 | #ColisSuspect au 51 avenue Marcel Cachin à @VilleDeSMH. |
| 24 | Une grande colonne de fumée noire est visible depuis la gare de @VilledePantin.#incendie |
| 25 | inondation à Mondeville en bas de la route de Cabourg.Amis cyclistes, pensez à lever les pieds.@VilleMondeville |
| 26 | @VilledeChallansQuand se passera t il quelque-chose contre ces incivilités ?Le jet de mégot est un dépôt d ordure sauvage. |
| 27 | @villepoissy @78actu Pourquoi il y a eu de la bagarre sur le parvis de la mairie |

TABLEAU 4.14 – Exemple de tweets pour chaque événement

# Troisième partie

# Identification d'événement

# Chapitre 5

# Ontologie événementielle

## Sommaire


5.1 Modèle ontologique . . . . . 120
5.2 Ontologie GEOLSemantics des personnes . . . . . 123
5.3 Ontologies gérant des événements . . . . . 126
5.4 Détection d'événements dans les tweets . . . . . 127
5.5 Représentation des événements dans l'ontologie GEOLSemantics . . . . . 133
5.6 Conclusion . . . . . 146


L'utilisation des connaissances en informatique est censée établir un dialogue entre l'homme et la machine et promouvoir un système d'aide à la décision. Pour cela, le système doit avoir accès aux termes utilisés par l'être humain et à leur sémantique. En ce sens, les ontologies nous permettent d'avoir un système interprétable à la fois par l'humain et par la machine [Hernandez, 2005].

Dans le contexte de cette étude, nous devons pouvoir corréler les événements avec les alertes de façon pertinente. Pour cela, il faut que le modèle soit porteur de sens. La syntaxe seule ne suffit pas. En effet, nous devons regrouper un ensemble d'informations filtrées depuis des tweets qui vont constituer l'événement même dont les connaissances, explicites et sous-jacentes, vont engendrer une alerte en résonance.

De ce fait, nous nous sommes intéressés aux ontologies événementielles pour lesquelles l'inférence de nouvelles connaissances et la cohérence des événements sont réalisées par un raisonneur.

Nos événements étant liés à des personnes et induits par l'intermédiaire de tweets (regroupés ou non), dans ce qui suit nous allons d'abord présenter la notion d'ontologie au sens large dans la section 5.1 ; puis nous décrirons l'ontologie pré-existante utilisée par GEOLSemantics qui permet de gérer des entités nommées mais pour laquelle nous nous focaliserons sur la représentation des personnes. Par la suite, nous présenterons dans la section 5.3, les ontologies événementielles existantes qui ont semblé intéressantes dans le contexte de notre étude. À partir de là, nous décrirons dans la section 5.5, les concepts spécifiques ajoutés dans notre ontologie pour représenter notre notion d'événement dans le contexte des réseaux sociaux.

## 5.1 Modèle ontologique

En 1993, [Gruber, 1993] introduit la notion d'ontologie comme étant une « spécification explicite d'une conceptualisation », définition qui, en 1997, sera légèrement modifiée par [Borst and Borst, 1997] en « une spécification explicite et formelle d'une conceptualisation partagée ». Une ontologie permet une communication entre l'homme et la machine mais aussi entre machines en définissant le sens des objets avec une représentation structurée de leur rôle [Studer *et al.*, 1998].

Les ontologies sont au cœur du web sémantique [Berners-Lee *et al.*, 2006]. Elles sont utilisées dans de nombreux domaines tels que l'ingénierie des connaissances [Allemang *et al.*, 2020], la modélisation qualitative, l'ingénierie des langages, la conception de bases de connaissances, la recherche d'information, l'extraction d'information, la gestion et l'organisation de connaissances, et plus récemment dans l'e-commerce [Guarino, 1998] et les infra-structures cloud pour l'internet des objets [1].

Une ontologie est un ensemble structuré de concepts. Selon Gandon and Dieng [2001], « un

1. Web of Things (WoT) Architecture

concept est un constituant de la pensée, né de l'esprit (un principe, une idée, une notion abstraite) sémantiquement évaluable et communicable ». Chacun de ces concepts peut-être caractérisé par des attributs (dits *data property*). Ces concepts sont organisés soit par une notion de relation hiérarchique (on parle alors de subsomption) correspondant à une sorte de taxonomie, soit par des relations sémantiques ou rôles (dite *object property*) ; par exemple, la relation *conduire* relie le concept de *Pilote* avec le concept de *Véhicule*. Pour ce genre de relation, nous pouvons définir un ensemble de propriété mathématiques. Toutes ces caractéristiques seront employées afin de raisonner à propos des objets d'un domaine [Alaya Mili, 2016]. De façon succincte, le raisonnement est le fait de tirer des conclusions plausibles sur la base d'éléments non suffisamment concluant en l'absence d'évidence d'information contraire [Rosier, 2015].

Plus précisément, une ontologie est constituée d'une terminologie (TBox) qui représente les concepts du domaine que l'on veut formaliser et d'un ensemble d'assertions (ABox). C'est une interprétation du monde réel constituant une base de connaissances [Noy and McGuinness, 2000] à partir de laquelle des raisonneurs comme HermiT [2] ou autres [3] peuvent vérifier la cohérence de cette interprétation et inférer ou non de nouvelles connaissances. Selon les constructeurs de la logique de description utilisés [4], la sémantique de l'ontologie sera plus ou moins grande.

Les ontologies peuvent donc être construites pour différents niveaux d'expressivité sémantique, en utilisant différents langages [Palmer, 2001] qui font partie de la famille du W3C [5]. M. Fennell [Fennell, 2013] considère que :

**XML** (pour *eXtensible Mark-up Language*) : est un langage de balisage qui définit un ensemble de règles pour coder des documents dans un format qui est à la fois lisible par l'homme et par la machine. XML est un format de données textuelles avec un support solide via Unicode pour différentes langues. Il est largement utilisé pour la représentation de structures de données arbitraires telles que celles utilisées dans les services Web ;

2. `http://www.hermit-reasoner.com/`
3. voir raisonneurs référencés par le W3C
4. pour plus de détails, voir la page de Pierre-Antoine Champin
5. World Wide Web Consortium

**RDF** (*Resource Description Framework*) : est utilisé comme méthode générale de description conceptuelle ou de modélisation des informations implémentées dans les ressources Web en utilisant une variété de notations de syntaxe et de formats de sérialisation de données.

**OWL** (*Web Ontology Langage*) : est une famille de langages de représentation des connaissances pour la création d'ontologies. Les langages OWL sont caractérisés par une sémantique formelle qui est l'étude de la sémantique ou l'interprétation des langages formels et naturels.

Dans le contexte du langage naturel et de son traitement, les informations sont souvent traitées comme l'entrée du processus de construction de l'ontologie, qui analyse le texte en noms et verbes. Les noms sont représentés comme des « classes » (ou concepts) et les verbes comme des « propriétés » associant des valeurs (*"data property"*) ou des relations entre classes avec certaines contraintes (*"object property"*).

Les ontologies peuvent être créées manuellement à partir de rien par l'intermédiaire d'éditeur comme Protégé[6] éventuellement guidés par des méthodologies comme NeOn[7] ou Methontology [Fernández-López *et al.*, 1997]. Elles peuvent aussi être construites en extrayant des informations du Web ou, éventuellement, en intégrant des ontologies déjà existantes. Mais ce processus manuel devient parfois très complexe et coûteux en temps, surtout lorsqu'il s'agit de traiter une grande quantité de données. Dans ce cas, les approches d'apprentissage automatique peuvent être très utiles pour générer automatiquement des ontologies car elles fournissent des schémas en temps réel. Parmi ces techniques, citons les règles de classification, le clustering, etc. (présentées dans la section 2.3.1).

Dans notre étude, nous avons étendu l'ontologie de GEOLSemantics que nous allons présenter maintenant.

6. `https://protege.stanford.edu/`
7. http ://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/methodologies/59-neon-methodology/

## 5.2 Ontologie GEOLSemantics des personnes

GEOLSemantics possède déjà une ontologie, fruit de trois projets de recherche sur l'incertitude [Kerdjoudj, 2015], le chinois [Wang, 2016] et l'arabe dialectal [Saâdane, 2015].

La première version de l'ontologie [Wang, 2016] a été créée dans le cadre du projet *SAIMSI*, et concernait les personnes, les organisations (cf. la représentation partielle de la figure 5.2) et la sécurité donc les concepts d'acte de violence, les procédures judiciaires (interpellation, arrestation, emprisonnement, ...), etc. La solution développée permet de suivre des activités de personnes qui sont soupçonnées d'activités illicites comme du terrorisme, de la vente de drogue, ou du blanchiment d'argent, etc.) à partir de sources textuelles ou vocales libres sur internet, et ceci en plusieurs langues (français, anglais, arabe et chinois) en traitant des données à la volée. Cette solution répond aux questions :

— qui a fait quoi,

— quand,

— où,

— comment,

— avec qui,

— en quelle quantité.

Dans le cadre de cette thèse, l'objectif sera d'optimiser l'ontologie de GEOLSemantics et de l'améliorer en s'inspirant des ontologies événementielles qui existent déjà et éventuellement rajouter des concepts liés aux actions d'alerte.

L'ontologie de GEOLSemantics est orientée activité des personnes. L'extraction des objets *"PhysicalEntity"* et des agents *"NamedEntity"* est déjà prévue par l'ontologie de GEOLSemantics (voir la figure 5.1).

Les concepts correspondants sont hiérarchisés selon le contexte d'utilisation et un certain nombres de relations sémantiques ont été définies comme on peut le constater dans la partie droite de la figure 5.1 où sont visibles quelques uns des rôles possibles entre personnes.

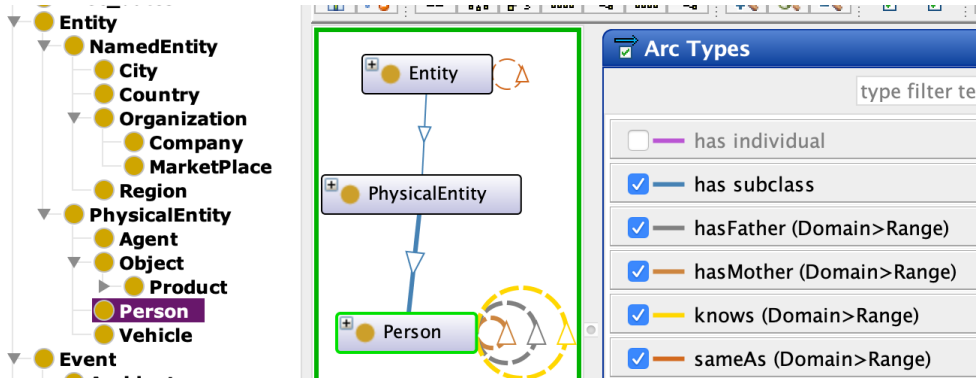



FIGURE 5.1 – Représentation partielle des entités

### 5.2.1 Intégration d'autres ontologies

Beaucoup de recherches abordent le problème de l'intégration de différentes ontologies. Michel Klein dans [Klein, 2001] propose une classification de différentes tâches et problèmes liés à la combinaison de différentes ontologies.

Quand on veut utiliser différentes ontologies ensemble, ces ontologies doivent être combinées d'une manière ou d'une autre. Cela peut être fait par :

— combinaison : utilisation de deux ou plusieurs ontologies différentes pour une tâche dans laquelle leur relation mutuelle est pertinente.

— fusion, intégration : création d'une nouvelle ontologie à partir de deux ou plusieurs ontologies existantes avec des parties qui se chevauchent, qui peuvent être virtuelles ou physiques.

— alignement : amenez deux ou plusieurs ontologies en accord mutuel, en les rendant cohérentes et consistantes.

— mapping : relier des concepts ou des relations similaires (selon certaines métriques) de différentes sources entre eux par une relation d'équivalence. Un mappage aboutit à une intégration virtuelle.

La technique adoptée par GEOLSemantics est une combinaison entre fusion et intégration. Les ontologies les plus utilisées sont en sus de celle propre à notre environnement sont vcard[8] et foaf[9] pour caractériser les entités (nommés et personnes), icaltzd[10] pour les

8. http://www.w3.org/2006/vcard/ns#
9. http://xmlns.com/foaf/0.1/
10. http://www.w3.org/2002/12/cal/icaltzd#

aspects temporels calendaires et GEO [11] pour les aspects géographiques.

Nous nous en sommes inspirés. Cependant, nous avons dû les transformer afin d'homogénéiser les propriétés dans notre ontologie, et surtout, afin de répondre au mieux au contexte de l'étude. Par exemple, une personne peut avoir un nom de famille, mais aussi un nom de jeune fille, ...

Dans ce travail, nous avons développé la description des actions afin d'en ajouter de nouvelles, en nous basant sur les besoins exprimés lors le projet *Safecity*. Ces actions apparaissent comme des événements particuliers (cf. figure 5.2).

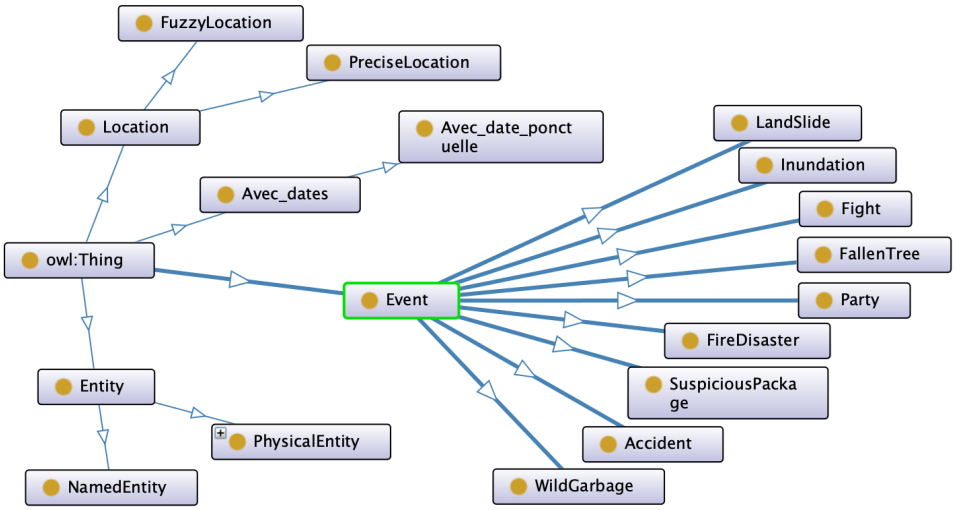



FIGURE 5.2 – Hiérarchie partielle des concepts pour les événements

Le concept d'événement étant considéré comme un triplet (action, lieu, date), la hiérarchisation des événements en action nous permet de faire une première classification.

Nous avons aussi développé la partie représentation des lieux en y intégrant les données GPS. Un lien vers OpenStreetMap et/ou Geonames a été ajouté pour faciliter par la suite la visualisation des localisations géographiques.

### 5.2.2 Limites pour la gestion d'événement

Dans le cadre de cette étude, la représentation d'un événement en triplet (action, lieu, date) qui devrait être déterminé notamment dans ses dimensions temporelles et géographiques

11. `http://www.w3.org/2003/01/geo/wgs84_pos#`

par le traitement linguistique des tweets est limité par le fait même que les tweets ne sont pas représentés dans l'ontologie. Il n'est donc pas possible pour l'instant d'inférer de la connaissance "événementielle" à partir des tweets. Par exemple, si nous collections deux tweets énonçant l'un *incendie à Nice* et l'autre *il y a le feu dans le nord de Nice* émis à 30 secondes d'intervalles, on devrait pouvoir établir un lien de corrélation entre les deux, inférer une instance dans la classe *FireDisaster* avec le rôle *hasLocation* vers une instance de la classe *Location* avec la date du tweet le plus ancien sachant que l'on aurait une traçabilité de l'évolution de l'événement au fur et à mesure de l'intégration d'autres tweets sur cette action. D'autre part, cela devrait détecter si deux événements sont au final identique.

## 5.3 Ontologies gérant des événements

Pour rechercher de l'information dans un corpus tout en respectant la notion de contexte et avoir un système à la fois interprétable par l'homme et par la machine, il nous a semblé adéquat de nous tourner vers la notion d'ontologie. Par exemple, le raisonnement va permettre de dire que deux événements de même type qui sont proches dans le temps et dans l'espace sont probablement les mêmes. Il va permettre de regrouper les informations sur un même événement, en conservant les informations les plus précises grâce à la construction de l'ontologie, mais aussi représenter les liens entre événements.

Pour les besoins de notre recherche, nous allons nous intéresser aux ontologies événementielles.

Si l'on considère qu'un événement est une action qui a lieu à un endroit particulier pendant une certaine période de temps, on trouve ici trois éléments importants à savoir une action, une position dans l'espace et une position dans le temps, qui en se combinant vont représenter un événement.

De nombreux chercheurs se sont intéressés aux ontologies événementielles dans des buts précis comme des chronologies historiques. Citons par exemple :

— LODE [Troncy *et al.*, 2010], une ontologie pour représenter des événements dans le

web de données ;

— CIDOC-CRM [Doerr, 2003] et ABC [Lagoze and Hunter, 2001] visent des événements historiques au sens large (*e.g.* guerre, naissance) ou liés aux objets décrits (*e.g.* changement de propriétaire, restauration), en manipulant des contenus multimedia possédés par les musées et les bibliothèques ;

— SEM permet d'identifier les événements en relation avec des attaques maritimes [Van Hage *et al.*, 2009]. C'est un modèle qui permet de d'écrire des événements passés, présents et futurs dans la presse ([Scherp and Mezaris, 2014]) ;

— DOLCE+DnS Ultralite (DUL) est une ontologie qui combine une axiomatisation simplifiée des concepts de DOLCE avec les patrons de représentation "*Descriptions and Situations*" [Gangemi and Mika, 2003]. Dolce [12] [Masolo *et al.*, 2003] est une ontologie fondamentale développée dans une perspective pluraliste qui est : il ne peut y avoir de norme unique ou d'ontologie universelle pour la représentation des connaissances. Dans Dolce, on fait la distinction entre l'objet et l'événement. Un objet est persistent (dure dans le temps) et l'événement est non persistant (ne perdure pas dans le temps). Selon Dolce aussi, la localisation d'un événement est celle de la localisation de ses participants [Borgo and Masolo, 2010] ;

— F-Event est fondée sur DUL dans la mesure où elle spécifie d'autres patrons pour représenter des relations entre événements, des relations méréologiques, causales, de participation et de corrélation [Scherp *et al.*, 2009] ;

— OpenCYC fournit des centaines de milliers de classes ayant pour but de modéliser « toute la réalité de l'homme » [Sicilia *et al.*, 2004].

## 5.4 Détection d'événements dans les tweets

Rappelons que Twitter est un site de microblogging, un micro-monde à lui seul [Pélissier and Gallezot, 2013], qui depuis sa création a suscité un grand intérêt de la part de la communauté scientifique, du fait de sa capacité à diffuser en temps réel les opinions, les

12. *Descriptive Ontology for Linguistic and Cognitive Engineering*

réactions et les contributions des utilisateurs, sur une information dans un premier temps, et leur propagation dans un deuxième temps. Il a permis, entre autres, de démocratiser la diffusion de l'information [Khamis *et al.*, 2017] ; tout le monde peut créer, compléter et propager une information, même si cela comporte un risque de propagation de fausses nouvelles [Guibon *et al.*, 2019].

Twitter possède un lexique spécialisé. Ce dernier est instable, notamment à cause du nombre restreint de caractères (140 signes passés à 280 depuis novembre 2017) qui pousse souvent à codifier par des contractions de mots, des acronymes, des émoticones, etc. Au commencement de Twitter, les travaux francophones se concentrant sur les forums de discussion, les courriels, les SMS, les chats ou les espaces d'apprentissage étaient plutôt rares [Paveau, 2013]. Pour le traitement linguistique des informations en ligne, nous pouvons citer [Pierozak, 2007], [Liénard and Zlitni, 2011] et [Frame and Brachotte, 2013] qui montrent la modification progressive des objets de la communication médiatisée par ordinateur (CMO) et la prise en compte récente des réseaux sociaux numériques (RSN) ([Develotte and Paveau, 2017]).

Twitter sert aussi à partager des événements dont les personnes sont témoins. C'est une plate-forme importante pour les professionnels de l'information tels que les journalistes. Une étude menée par [McGregor and Molyneux, 2020] a révélé que ceux qui utilisent moins Twitter dans leur travail ne tiennent pas compte de certaines actualités qui apparaissent sur la plateforme, ce qui les amène potentiellement à rejeter des informations que beaucoup de leurs collègues considèrent comme dignes d'intérêt car elles sont retweetées ou taggées par les *tweetos*[13].

La détection d'événements dans les tweets est une problématique longtemps étudiée et dans de nombreux domaines ; par exemple, détection des tremblements de terre [Sakaki *et al.*, 2012], détection d'épidémie de grippe [Lee *et al.*, 2013], détection d'événements financiers [Pagolu *et al.*, 2016], etc. Plusieurs méthodes sont utilisées pour cela.

Ansah *et al.* [2020] ont essayé de détecter les événements de protestation en utilisant les médias sociaux afin d'alerter les services d'urgence grâce à un système qui suit la

13. membre de la communauté Twitter

propagation des informations dans les communautés du réseau Twitter, en particulier le changement soudain dans le nombre d'interactions sur un tweet ou un mot-clé donné (ou *burst detection*).

Récemment, Mazoyer *et al.* dans [Mazoyer *et al.*, 2020] ont présenté un corpus français de tweets annotés pour la détection d'événements sur lequel ils ont effectué des expérimentations en utilisant des approches allant de l'utilisation de vecteurs tf-idf simples jusqu'aux techniques d'incorporation plus récentes fondées sur les réseaux de neurones.

À GEOLSemantics, nous avons abordé cette problématique en définissant d'abord l'événement par un triplet $< action, lieu, date >$. Chaque élément du triplet est d'abord extrait en utilisant un système d'extraction d'information basé sur une ontologie. Ce système nous a permis d'établir des règles afin de :

1. regrouper des tweets qui parlent du même événement améliorant la précision de l'événement et donc la pertinence de l'alerte ;
2. regrouper plusieurs événements. Par exemple, dans *chute d'arbre qui provoque une route bloquée*, deux événements apparaissent à savoir la chute d'un arbre et le fait qu'une route soit bloquée. Comme il y a conséquence d'un événement par rapport à un autre, une fusion sera effectuée.

Ceci va nous permettre de suivre l'événement dans le temps mais aussi de calculer sa véracité en fonction du nombre et du type des personnes qui tweetent sur l'événement.

Les deux problèmes majeurs rencontrés lors de l'extraction d'événements à partir de tweets sont :

— l'imprécision temporelle (la date/heure de l'événement n'est pas mentionnée systématiquement ou des expressions de date/heure relatives sont utilisées comme « vient de », « va organiser », « demain », etc.) et

— l'imprécision géographique (« pas loin de », « devant », etc.).

Concernant la détection des dates dans les tweets, par exemple, pour la prédiction d'événements, en se fondant sur une approche d'analyse sémantique automatique et sur des traitements NLP [14] sur des tweets, et en utilisant une modélisation linéaire, Wang *et al.*

14. Natural Language Processing

[2012] ont pu prédire des délits de fuite. Aramaki *et al.* [2011] ont pu démontrer que des instruments NLP peuvent permettre l'extraction d'informations pertinentes, en utilisant un classifieur SVM [15] sur un corpus de tweets sur la grippe. Sakaki *et al.* [2010] ont pu détecter en temps réel des tremblements de terre et lancer une alerte en avertissant des utilisateurs enregistrés, alerte émise bien avant celle de la JMA [16].

La détection des lieux, quant à elle, se fait sur quelques paramètres à savoir le tweet de l'utilisateur, son réseau (*followers*, *following*) et ses métadonnées (*profile location*, *tweet timezone*) [Mourad *et al.*, 2019].

Selon Hecht *et al.* [2011], 34% des lieux déclarés sont des lieux sarcastiques. Twitter permet aux utilisateurs d'afficher leurs coordonnées GPS ; mais seulement 1% des utilisateurs le font vraiment selon Han *et al.* [2013].

Nous avons vérifié cet état de fait sur environ 1000 tweets extraits sur la région parisienne par *bounding box*, c'est-à-dire que les utilisateurs permettent à l'application de se connecter à leur GPS. Ce test a révélé que 19% des tweets n'avaient pas déclaré de lieu dans *profile location*, 38% ont déclaré un autre lieu, 16% ont déclaré un lieu inexistant (cas des *troll*) et seulement 28% ont déclaré la bonne localisation, des enseignes de commerce pour la plupart. D'autre part, nous avons aussi réalisé un autre test en extrayant 1000 tweets commençant par *« Je suis à Paris »*. Dans ce cas, 38% des utilisateurs n'avaient pas déclaré le lieu, 35% avaient déclaré un autre lieu, 20% étaient des trolls et seulement 8% avaient déclaré la bonne géolocalisation Paris.

Le lieu déclaré dans le tweet est considéré comme une valeur sûre pour la détection du lieu par rapport à la géolocalisation de Twitter.

Dernièrement, Huang and Carley [2019] a proposé une géolocalisation hiérarchique pour des utilisateurs de Twitter, fondée sur un réseau de neurones.

Parmi les travaux réalisés sur les tweets, citons la détection des lieux dont le but est d'identifier les variations lexicales géographiques [Eisenstein *et al.*, 2010], Ces deux détections sont essentielles pour détecter des événements, notamment lors de la gestion des

15. Support Vector Machine
16. Japan Meteorological Agency

catastrophes naturelles [Kryvasheyeu *et al.*, 2015] ou le suivi d'une épidémie [Broniatowski *et al.*, 2013] entre autres. Pour Hoang and Mothe [2018], les événements sont représentés par trois dimensions principales : le temps, le lieu et les informations relatives à l'entité. Ils ont fait des travaux sur la relation rappel/précision, combinant différentes méthodes pour extraire les lieux. En ce qui nous concerne, nous visons essentiellement à identifier les événements contenant au minimum une indication de lieu et une temporalité au présent ou au passé proche, exprimés dans le contenu du message, afin de lancer des alertes. Les informations relatives à l'événement permettent de le caractériser par des précisions sur qui (personne ou organisation) fait quoi (action, événement, expérience), quand (date), où (lieu), avec quoi (objet) et combien (quantité ou mesure). Ceci devrait permettre de lever l'ambiguïté des entités et des actions grâce à la définition des contextes sémantiques.

L'extraction d'information depuis Twitter notamment la détection des événements et leur localisation sur l'espace-temps est donc un défi que la communauté scientifique essaie de relever depuis l'avènement des réseaux sociaux. Nous pouvons diviser les algorithmes de classification en deux catégories à savoir les méthodes orientés documents (*documents pivot methods*) et les méthodes orientées caractéristiques (*feature-pivot methods*). La première approche permet de détecter l'événement en utilisant du clustering [Yang *et al.*, 1998] tandis que la deuxième est plus sémantique car elle étudie la distribution des mots et extrait des événements selon un motif de mots définis [Kleinberg, 2003]. En suivant les méthodes orientées caractéristiques (*feature-pivot methods*) qui sont plus sémantiques car elles étudient la distribution des mots et extraient des événements selon un motif de mots défini (dit *pattern*) [Kleinberg, 2003], un événement précis pourrait être défini. Un événement est donc représenté par un certain nombre de mots-clés avec un nombre d'apparition [Weng and Lee, 2011]. Nous pouvons aussi observer que la détection d'un événement dans un lieu et une date précis nous incite à avoir une micro-détection de la date, du lieu et de l'action (selon 1. de 5.4) avant d'enchaîner sur une macro-détection des événements (selon 2. de 5.4).

Nous allons maintenant aborder le processus qui permet d'avoir ces visions micro et macro d'événements.

La figure 5.3 présente les traitements successifs induits par notre approche constitués de

trois parties : (i) la collecte des tweets et l'extraction d'information en deux dimensions (partie de gauche en bleu), (ii) la chaîne de traitement GEOL en quatre étapes (partie centrale en jaune) et (iii) la sortie de l'événement formalisé pour l'envoi d'alerte en JSON (partie de droite en rouge) dont le détail est donné dans les sous-sections suivantes.

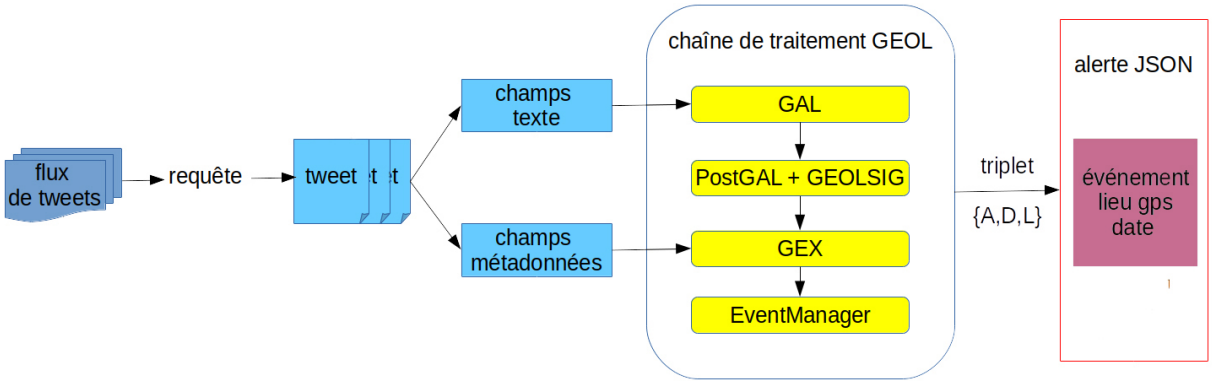



FIGURE 5.3 – Processus complet de détection d'événements géo-chrono-localisés

### 5.4.1 Regroupement des tweets d'un événement

Trois éléments doivent être présents pour définir un événement $E$ pour lancer une alerte, à savoir $A = action, L = Lieu, D = Date$. Ces trois éléments vont nous permettre de regrouper des tweets parlant du même événement, et ce, même si dans certains tweets, tous les éléments ne sont pas renseignés. Il s'agit donc d'obtenir : $E = (A, L, D)$ où $A = f_{action}(\{tweet\}_n)$, $L = f_{lieu}(\{tweet\}_n)$, $D = f_{date}(\{tweet\}_n)$ où $f$ représente la fonction de raffinement d'un des critères *action*, *lieu*, *date*. Notons que *lieu* peut être un lieu précis ou une zone géographique, et *date* une date précise ou un intervalle de temps.

Une fois le pré-traitement réalisé, lors de la phase d'extraction, nous avons cherché à repérer des événements composés d'une action, d'un lieu et d'une date. Nous avons vu précédemment que les tweets sont des textes courts, mal rédigés et avares en informations. Par exemple, la date est rarement mentionnée lorsque l'événement est en cours. Aussi, un certain nombre d'adaptation des traitements linguistiques couramment employés sur des documents textuels sont nécessaires. L'extraction des événements repose sur une analyse linguistique profonde (cf. module GAL de la figure 5.3) et une extraction qui permet de repérer des entités nommées, soit personnes, organisation, lieu, date, etc., (cf. module

GEX dans la figure 5.3) grâce à l'ontologie propre à GEOLSemantics (cf. section 5.2) et un module d'amélioration de la reconnaissance des lieux.

## 5.5 Représentation des événements dans l'ontologie GEOLSemantics

L'ontologie de GEOLSemantics est orientée activités des personnes. La recherche de localisation d'événement actuellement réalisée est une vision assez différente de la représentation sémantique, y compris d'événement, déjà existante dans l'ontologie. Par exemple, l'événement « acte de violence » représenté dans la précédente ontologie est surtout orienté pour connaître tous les actes de violence réalisés par une personne identifiée. Dans le cas du projet *SafeCity*, il est évident qu'il y a peu de chance de détecter la personne agissante ou les victimes. Dans le cas présent, ce qui intéresse les autorités, c'est le type d'incident et son lieu précis. Ensuite, certaines précisions peuvent être utiles comme le type d'acte commis (par exemple, *coup de feu, bombe, vol, etc.*), le genre de l'auteur, le nombre de blessés, l'instrument de l'acte commis s'il est connu.

En ce qui concerne les autres types d'événements, tout a dû être décrit toujours dans l'optique de donner aux autorités le maximum d'information pour préparer leurs actions/-réponses, tout en s'appuyant sur les comportements de rédaction constatés dans les corpus récoltés.

Rappelons que tout événement n'est pas à signaler comme donnant lieu à une alerte. Il doit s'agir d'un événement qui se déroule dans le présent. La séparation entre événement actuel et événement passé doit faire l'objet d'une étude sur les corpus rassemblés. Notons cependant, qu'à la demande des autorités de Nice, un type d'événement futur doit être aussi identifié. Il s'agit de l'organisation de fêtes ouvertes aux internautes et qui peuvent rassembler un grand nombre de personnes.

La reconnaissance des événements est représentée par une ontologie décrivant chacun d'eux avec ses différents paramètres. Par exemple, un accident de la route sera représenté par la date, voire l'heure, le lieu, le type de véhicule, la présence de blessés ou non, etc.

En s'appuyant sur le résultat de l'analyse morphosyntaxique, une extraction de connaissances est réalisée par des règle. Celles-ci se déclenchent pour remplir les concepts $A, L, D$ définissant l'événement tels que décrits dans l'ontologie.

Dans le cas de textes longs, l'analyse morphosyntaxique permet d'identifier des entités identiques même si elles ne sont pas écrites de la même façon lors d'une dernière étape de traitement. Cette étape n'est pas utile dans le cas des tweets. En revanche, une levée d'ambiguïté sera réalisée en confrontant les lieux mentionnés dans le texte avec les lieux trouvés dans les métadonnées mais aussi les lieux connus dans une base de données géographique locale. Cette base de données géographique locale a été créée à partir de fichiers de l'*open data* des collectivités locales [17] et une extraction locale de fichiers de l'*open data* global notamment *Openstreetmap* [18] (Plus de détails sur la constitution des ressources dans la section 6.2.2). L'information ainsi extraite est chargée dans une base indexée par Elasticsearch [19] qui permet de rechercher les entités géographiques semblant les plus pertinentes par rapport à celles repérées dans le texte. Une fois choisie l'interprétation la plus proche, le système retourne les coordonnées de l'entité géographique contenue dans le texte.

Dans un premier temps, le travail a porté principalement sur la modélisation de l'ontologie établissant une taxonomie des concepts relatifs à tous les événements mais aussi leurs relations et leurs propriétés dont les notions temporelle et spatiale. Des axiomes sont définis pour augmenter la sémantique de la taxonomie.

### 5.5.1 Concepts issus de *SafeCity*

Pour chaque événement imposé par le projet *SafeCity*, nous devons extraire les propriétés d'action, de temps, d'objet, de lieu, d'agent, de cause, de conséquence, de type et de date selon l'événement (cf. tableau 5.3).

Comme mentionné précédemment, nous avons décidé de faire évoluer l'ontologie existante de GEOLSemantics pour le projet *SafeCity* en la modifiant en conséquence grâce à notre

17. `https://www.data.gouv.fr/fr`
18. `http://openstreetmap.fr`
19. `https://www.elastic.co/fr/products/elasticsearch`

processus de recherche d'information à partir de tweets. En effet, il faut rester sémantiquement proche de l'ontologie GEOLSemantics tout en ayant une certaine liberté dans la modélisation des concepts afin de pouvoir les intégrer avec ceux de l'ontologie GEOLSemantics initiale pour garder sa cohérence.

Notre première approche a été de créer trois nouvelles classes, classe événementielle *"Event"*, classe de localisation *"Localisation"* et classe temporelle *"Avec_dates"*.

Pour les besoin du projet *Safecity*, nous sommes partie de rien. Nous avons collecté des tweets pertinents sur chaque événement attendu et défini les concepts ainsi que les règles d'extraction pour chaque élément pertinent relatif à la date, le lieu et le type de l'événement.

### 5.5.2 Classe événementielle

Une première réflexion sur l'ontologie (cf. section 5.2) nous a amené à transformer chaque événement en concept avec des propriétés pour chaque concept.

Les événements sujets à déclenchement d'alerte sont classés selon :

**la durée** : afin de permettre le regroupement de tweets tant que la durée de l'événement est pertinente. Par exemple, si un tweet décrit un accident, dans les 5 heures suivantes, d'autres tweets sur le même accident, susceptibles de compléter ou de mettre à jour l'information du premier tweet, peuvent être postés. La classification des événements selon la durée est spécifiée dans le tableau 5.1 ;

| **Court (5h)** | **Moyen (12h)** | **Long/Futur (7jours)** |
|---|---|---|
| Colis suspect | Inondation | Organisation de fêtes |
| Accident | | Dépôt d'ordures sauvages |
| Incendie | | |
| Chute d'arbre | | |
| Bagarre | | |
| Glissement de terrain | | |

TABLEAU 5.1 – Liste des événements selon la durée

**- la priorité** : dans le cas de plusieurs événements dans le même tweet, l'alerte va se déclencher sur l'événement le plus prioritaire. Les événements sont divisés en catégories à savoir :

1. événements d'origine humaine graves ;
2. événements naturels ;
3. événements d'origine humaine moins graves.

| Événement | Catégorie | Priorité |
|---|---|---|
| Accident | 1 | 1 |
| Bagarre | | 2 |
| Colis Suspect | | 3 |
| Incendie | | 4 |
| Chute d'arbre | 2 | 5 |
| Inondation | | 6 |
| Glissement de terrain | | 7 |
| Organisation de fêtes | 3 | 8 |
| Dépôts d'ordures sauvages | | 9 |

TABLEAU 5.2 – Liste des événements par ordre de priorité

Dans chacune des catégories, les événements sont classés par ordre de gravité (cf. tableau 5.2). Dans le cas d'un tweet contenant plusieurs événements, un événement est signalé plus prioritaire. Selon la configuration, soit une seule alerte sera généré selon cette priorité (ceci est la configuration par défaut car c'est une contrainte de nos partenaires du projet Safecity), soit autant d'alertes que d'événements.

### 5.5.3 Classe de localisation

Comme nous venons de le voir, malgré la nature des tweets qui est limitée en taille, on peut avoir plusieurs événements et plusieurs lieux dans le même tweet. Nous avons vu que dans le cas des événements on renvoie l'alerte sur l'événement le plus prioritaire (cf. sous-section 5.2). En ce qui concerne les lieux ils seront traités comme suit :

Le lieu de l'événement peut être précis ou fixe (par exemple, *une adresse, une gare, un arrêt de bus*, etc.) comme il peut être approximatif (*vers ..., entre deux stations, entre deux sorties d'autoroutes*, etc.).

La représentation des lieux est une liste d'au moins un point, ce dernier composé d'une longitude et d'une latitude. La liste est appelée *geo-shape*[20]. Afin de construire la base de données du GEOLsig, nous avons pris en considération les geo-shapes suivants :

— **Point** : une coordonnée géographique unique, composée d'une longitude et d'une latitude [[lon,lat]] ;

— **Polygone** : un ensemble de points dont le premier et le dernier doivent correspondre [[lon1,lat1], [lon2,lat2], [lon3,lat3], [lon1,lat1]] ;

— **LineString** : une ligne arbitraire avec deux points ou plus [[lon1,lat1], [lon2,lat2], [lon3,lat3]] ;

— **Cercle** : un point unique avec un rayon [[lon,lat,rayon]].

Dans le cas des lieux précis, le lieu affiché sera exactement celui qui se trouve dans la base de données GEOLsig.

Pour les lieux approximatifs, les coordonnées renvoyées de la base de données GEOLsig seront modifiées avant l'affichage. Trois types de changement seront pris en considération selon :

— la combinaison des lieux : dans l'exemple « je suis entre Le méridien et le marché de Noël », les lieux renvoyés par la base de données seront une liste de points qui composent *Le méridien* et une autre qui composent *le marché de Noël*. Le lieu affiché sera un cercle composé de la moyenne des deux centroïdes[21] du marché de Noël et du méridien, et un rayon qui correspond à la distance entre le centroïde et le point le plus proche de tous les points qui composent les deux lieux ;

— le lieu modifié par une préposition : dans l'exemple « Je suis aux alentours de la tour smart », le lieu renvoyé par la base de données sera les coordonnées de la tour smart. Le lieu affiché sera le centroïde de la tour smart et un rayon qui double la distance

20. https ://www.elastic.co/guide/en/elasticsearch/reference/current/geo-shape.html#geo-shape
21. La moyenne des points qui composent le lieu [moyenne longitude, moyenne latitude]

entre le centroïde et le point le plus éloigné de ceux composant les coordonnées de la tour smart.

— un choix entre les lieux : dans certains cas, il faut renvoyer un des lieux, généralement le plus précis. Dans l'exemple, « Je suis à Paris, à place de la République », *Paris* va nous servir à renvoyer *place de la République* qui se trouve à Paris, et non pas celles qui se trouvent dans d'autre villes. Ici, le premier lieu est utilisé pour apporter de la précision et il est inutile de l'afficher.

Les détails du traitement des lieux flous sont décrits à la section 3.3.2.

La figure 5.4 montre l'affichage d'un lieu fixe, tandis que la figure 5.5 nous montre un cercle entre les deux lieux présents dans les tweets. À titre comparatif, la figure 5.6 montre qu'un des outils les plus utilisés dans la géolocalisation a choisi de renvoyer le lieu le plus précis au lieu de combiner, ou trancher selon la sémantique des prépositions.

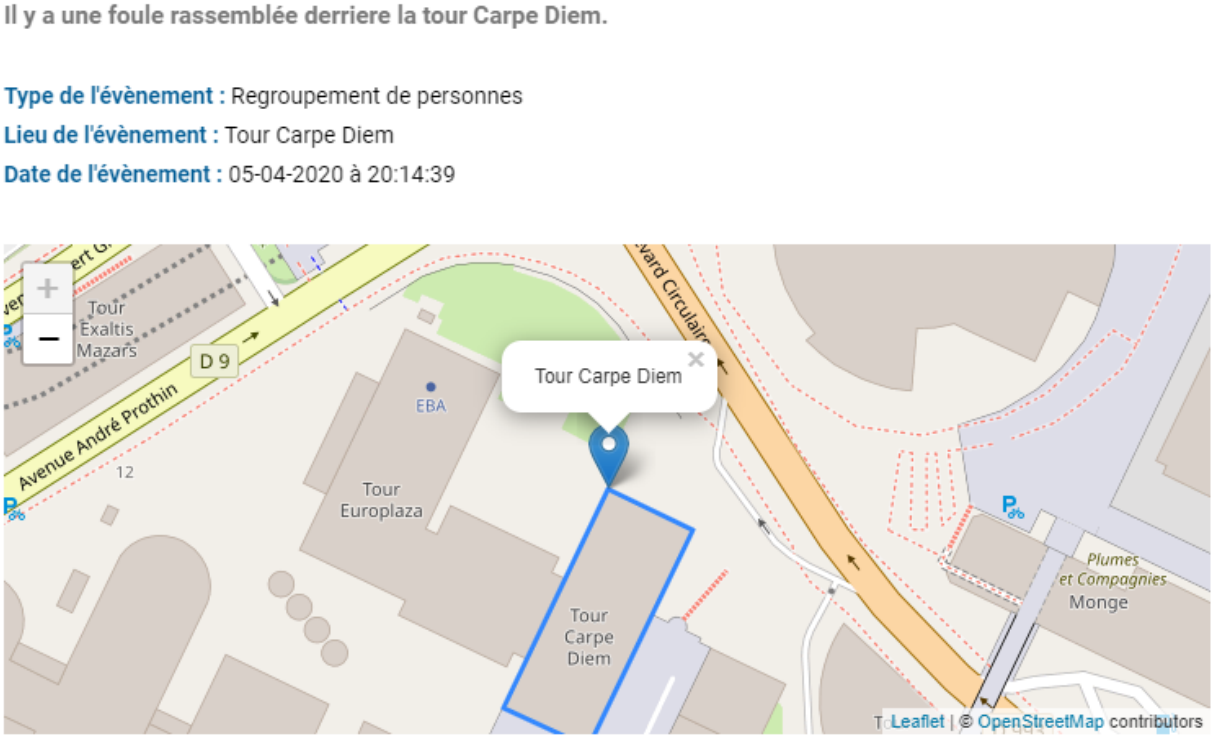



FIGURE 5.4 – Représentation d'un lieu fixe

Les propriétés du concept de lieu sont :

— name : chaîne de caractères désignant le nom du lieu . Par exemple, « La Baraca » ;

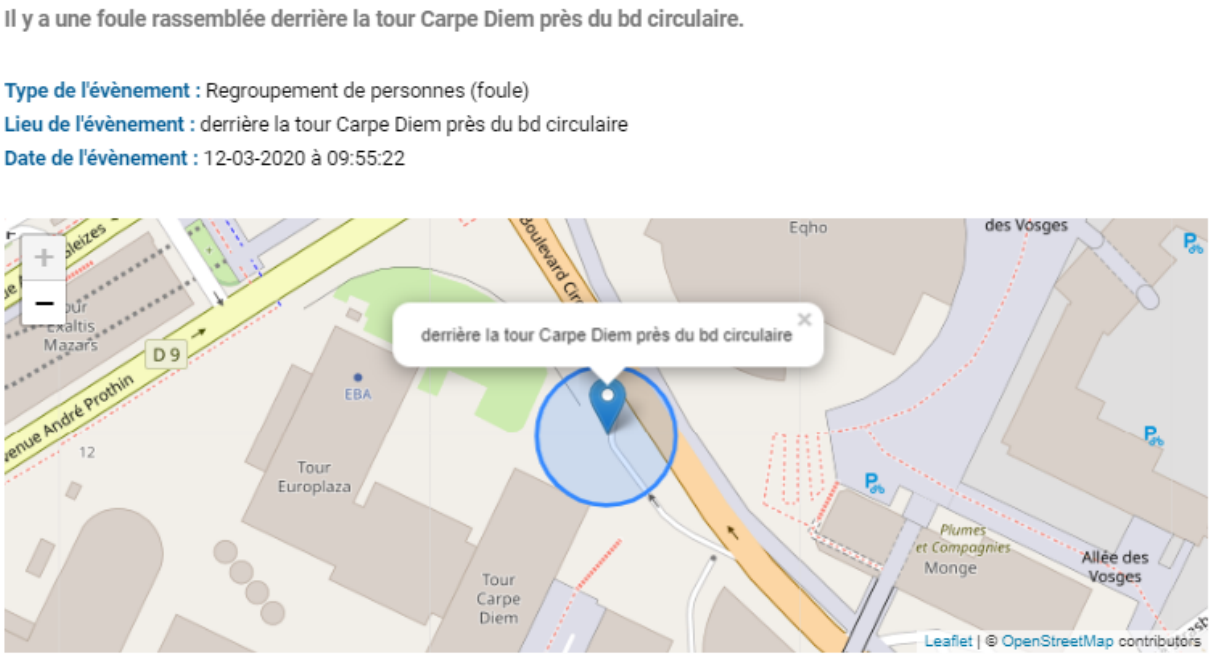



FIGURE 5.5 – Représentation d'un lieu flou

— location-type : chaîne de caractères désignant le type de lieu dont il s'agit : restaurant, rue, immeuble, quartier, etc. ;

— hasMesure : booléen qui sert à faire un lien vers une mesure qui permet de représenter par exemple la distance entre ce lieu et un autre ;

— Mesure : réel qui sert à définir la mesure entre deux lieux ;

— hasGeoPosition : booléen qui sert à informer si un lieu possède des coordonnées géo-spatiales ;

— coordinates : liste des longitudes et des latitudes représentant un lieu. Par exemple, [-0.02,51.4], (format geoJson [22]).

Les coordonnées seront utilisées par Elasticsearch6.2.2 grâce à un champs de type *geo_point* qui permettra ensuite de faire des recherches avancées sur celles-ci. Notons que *geo_point* peut être défini comme un objet (object), une chaîne de caractères (string) ou un tableau (array).

Notre formalisation se traduit concrètement sous forme RDF par une paire de coordonnées selon les deux exemples suivants.

22. https ://geojson.org

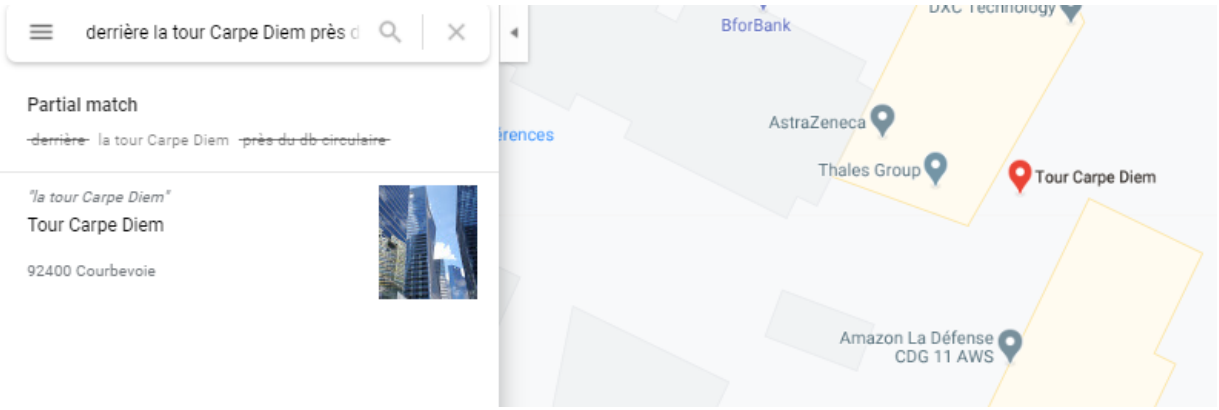



FIGURE 5.6 – Représentation d'un lieu flou avec une interface d'un outil tier

**Exemple 5.5.1.** Géolocalisation par un point

```
<v:Location rdf:nodeID="id0Location">
   <posBeg>0</posBeg>
   <gs:loc-type>ville</gs:loc-type>
   <posEnd>20</posEnd>
   <posBeg>0</posBeg>
   <gs:loc-name>Paris</gs:loc-name>
   <gs:hasGeoPosition rdf:nodeID="id0GeoPosition"/>
</v:Location>
...
<v:GeoPosition rdf:nodeID="id0GeoPosition">
   <gs:coordinates>[-0.02,51.4]<gs:coordinates>
</v:GeoPosition>
```

**Exemple 5.5.2.** Géolocalisation par un multipoint

```
<v:Location rdf:nodeID="id14Location">
 <gs:loc-type>ville</gs:loc-type>
 <gs:location-name rdf:ID="location-name_14_19id14Location">Paris</gs:location-name>
</v:Location>
<v:GeoPosition rdf:nodeID="id0GeoPosition">
  <gs:coordinates>
  [[7.2702742, 43.6956968],
  [7.2703426, 43.6956980],
  [7.2707322, 43.6957057]]
  </gs:coordinates>
</v:GeoPosition>
<rdf:Description rdf:about="#location-name_14_19id14Location">
  <gs:hasOcc>
    <gs:OccurrencePos>
      <gs:posBeg>0</gs:posBeg>
      <gs:posEnd>20</gs:posEnd>
    </gs:OccurrencePos>
  </gs:hasOcc>
</rdf:Description>
```

### 5.5.4 Classe temporelle

Dans la même logique que la représentation de la localisation, une date peut être ponctuelle ou correspondre à une fenêtre temporelle. Dans notre ontologie, nous représentons toute date comme un intervalle temporel avec deux propriétés *start_date* et *end_date* ; une date ponctuelle est alors une date où $start_date = end_date$.

Nous distinguons deux notions de dates pour un événement : la date de survenue de l'événement, et la durée de l'événement.

La durée d'un événement ne décrit pas quand il commence et quand il finit, mais dans quel intervalle il est raisonnable de penser que tout événement de même type se produisant au même endroit est le même événement, et non un événement distinct.

Voici les règles de regroupement d'événements que nous avons mis en place :

— tweets sur une même action avec proximité de lieu et de temps => regroupement ;

— tweets sur une même action avec proximité de lieu mais pas de temps => non regroupement ;

— tweets sur une même action avec proximité de temps mais pas de lieu => non regroupement ;

— tweets sur une même action avec aucune proximité de temps et de lieu => non regroupement

Chaque tweet avec événement peut être relié (ou non) à un autre tweet avec événement selon les proximités géographique de lieu et temporelle.

En effet, afin d'éviter de lancer plusieurs alertes pour un même événement, nous testons chaque tweet avec les tweets qui ont lancé une alerte précédemment. Si le nouveau tweet déclenche le même événement dans une période de temps relativement proche du tweet de l'ancienne alerte (la figure 5.1 montre la durée de regroupement des tweets) et à 20m (à titre expérimental, cela peut changer selon les exemples) du lieu du tweet de l'ancienne alerte, il est regroupé avec celle-ci afin de donner une indication sur l'importance de l'événement (nombre de tweets parlant de cet événement).

Cette relation ne peut pas forcément être résumée par une intersection des deux lieux d'événements, comme on peut le constater dans les cas de figure suivants.

### 5.5.5 Relations entre événements

Dans la même idée que des tweets sont regroupés car ils parlent du même événement, nous avons pris le parti de considérer que des événements peuvent être liés ou non selon certains critères que nous allons présenter maintenant.

**Événements liés**

- précision : l'événement 2 explique l'événement 1 et il est plus précis : l'intersection du (lieu1 et lieu2) = lieu1 ou vice-versa
  - tweet 1 à 9h :31m : foule qui se regroupe vers le marché de Noël
  - tweet 2 à 9h35m : bagarre à la porte 2 du marché de Noël
- déplacement : l'événement 2 montre que le lieu de l'alerte se déplace ; dans l'exemple suivant il faut signaler les deux lieux, le deuxième pour l'arrestation le premier pour aider la victime
  - tweet 1 à 9h :31m : Homme qui attaque une personne à Gentilly devant le RER
  - tweet 2 à 9h40m : Homme qui court avec un couteau devant la mairie de Gentilly
- Propagation : les tweets permettent de montrer la propagation d'un événement, bien souvent pour les incendies
  - tweet 1 à 9h :31m : Une fumée s'échappe de la tour carpe diem
  - tweet 2 à 10h40m : un incendie à la Tour EuroPlaza
  - |euroPlaza|carpeDIEM| sont deux immeubles côte à côte

**Événements non liés**

- tweet 1 à 9h :31m : Un accident à l'entrée de la ville
- tweet 2 à 10h40m : Une prise d'otages en court à la banque de l'entrée de la ville

— bien que les deux événements sont au même lieu au même moment, il n'y a a priori aucun rapport de cause à effet entre les deux et chacun a besoin d'une équipe d'intervention différente

Chaque cas nécessite un traitement particulier à savoir :

— précision : intersection de lieux ;

— déplacement : utilisation du lieu de l'événement le plus récent ;

— propagation : fusion des lieux des événements ;

— non lié : pas de regroupement d'événements.

**Gestion des éléments de liaison d'événements** Grâce à la sémantique sous-jacente à nos événements, des règles peuvent être établies selon les types des événements : les événements qui se propagent, les événements qui se déplacent et les événements qui apportent une précision contextuelle. D'autre part, la relation existante entre différents lieux est aussi un facteur décisif pour une meilleure localisation. En effet, des lieux qui partagent un point ou une ligne (combinaison de points) ne doivent pas être considérés de la même façon que des lieux dont l'un est inclus dans l'autre (précision). Le fait de garder les informations des tweets précédents est aussi un bon facteur de localisation.

Une métadonnée peut être ajoutée à chaque tweet déclencheur d'une alerte afin de garder la trace des combinaisons de déplacements et des propagations des événements précédents. Cela va permettre de constituer un *pattern* qui va servir à effectuer des prédictions sur l'état de l'événement. Le prochain tweet qui exprime cet état sera plus probablement regroupé.

**Exemple 5.5.3.** Soient les deux tweets

— Tweet 1 : Incendie à la tour Carpe Diem à la défence.

— Tweet 2 : Incendie à la tour Europe à la défence.

Les deux lieux étant proches l'un de l'autre, et le feu ayant comme nature de se propager, on va pouvoir regrouper les tweets 1 et 2.

**Exemple 5.5.4.** Soient les deux tweets

— Tweet 3 : Rave party devant la tour Carpe Diem à la défence.

— Tweet 4 : Rave party devant la tour Europe à la défence.

Les tweets 3 et 4 ne seront pas regroupés malgré la proximité géographique. Chaque tweet va servir à renvoyer une alerte, puisque l'événement organisation de fêtes ne possède pas la notion de propagation.

Il existe des contraintes dans la combinaison des relations. Par exemple, un déplacement peut être précisé mais une propagation ne peut pas l'être car cela n'a aucun sens. Il faut décider de la manière de représenter ces types de relation et débattre d'un éventuel lien avec les types d'événements. En effet, nous avons discuté précédemment de la possibilité de créer un dictionnaire associant pour chaque type d'événement la liste des événements avec lesquels il peut être combiné. Cette structure pourrait être améliorée pour ajouter en plus la relation associée à l'événement. Par exemple, on pourrait considérer qu'un incendie ne peut que se propager.

De nombreuses études ont été réalisées à propos de la nature d'un événement et de son interaction avec son environnement spatial. Selon Davidson [2001], les événements font partie intégrante de la métaphysique et de la machinerie sémantique d'interprétation des énoncés du langage naturel. La plupart des recherches sur la sémantique des événements depuis Parsons [1994] et Davidson [2001] se sont concentrées sur les questions relatives aux classifications aspectuelles ou à la sémantique temporelle. Eschenbach *et al.* [1999] a présenté la notion de « forme » d'un mouvement, et Pustejovsky and Moszkowicz [2011] a utilisé des verbes de mouvement comme prédicats de « création de chemin ».

Selon Pustejovsky [2016] la localisation d'un événement peut faire référence à l'un des quatre types de mouvements :

— transition : mouvement le long d'un chemin ;

— rotation : mouvement d'un corps autour d'un axe

— oscillation : mouvement périodique d'avant en arrière ;

— déformation : flexion, étirement et torsion d'un objet.

Pustejovsky [2013] explore également la sémantique du changement en donnant la structure d'événement suivante :

$evenement \rightarrow etat|processus|transition$

La théorie sémantique dynamique de l'interprétation du discours (*Dynamic semantic theory of discourse interpretation*) peut aussi être une alternative ou une amélioration du système de GEOLSemantics. Celle-ci utilise des relations rhétoriques pour modéliser l'exemple d'interface sémantique / pragmatique (ce qui est représentable dans une ontologie grâce à la logique de description sous-jacente).

Par exemple, « Un homme est entré. Il a commandé un verre » sera représenté par :

$$\exists homme(x) \Lambda entrer(x) \Lambda \exists verre(y) \Lambda commander(x, y)$$

Dans la sémantique dynamique, une phrase est interprétée comme une relation entre un contexte d'entrée et un contexte de sortie [Lascarides and Asher, 2008], ce qui peut être idéal dans un contexte de tweet, par nature court, regroupant un nombre limité d'objets, d'événements, de lieux, etc.

Jusqu'ici, le système de GEOLSemantics était utilisé pour le traitement de textes journalistiques ou juridiques qui sont, par nature, longs, avec l'utilisation d'anaphores parfois pointant sur un objet cité plusieurs lignes auparavant. C'est donc un contexte très différent des tweets.

Ceci dit, pour ce travail, nous allons nous placer dans le rôle d'utilisateur des ontologies. En ce qui concerne l'extraction de l'événement, nous allons utiliser le système actuel de GEOLSemantics basé sur une ontologie et des règles d'extraction afin d'extraire les événements. Un filtre sera rajouté afin de valider l'événement selon sa date, la présence d'une négation dans la phrase ou l'utilisation d'une métaphore comme nous allons le voir dans le chapitre 7.

### 5.5.6 Extension des classes GEOLSemantics

Pour l'extraction des objets et des agents, nous avons à notre disposition deux concepts, *physicalEntity* et *NamedEntity*, qui pré-existent dans l'ontologie de GEOLSemantics (cf. figure 5.1). Cependant, n'étant pas prévues à l'origine pour la gestion événementielle, nous avons mené une étude pour voir comment les utiliser de la façon la plus optimale.

| | Incendie | Accident | Arbre tombé | Inondation | Dépôts d'ordures | Colis suspect | Organisation de fêtes |
|---|---|---|---|---|---|---|---|
| Action | incendie, brûler, en feu... | accident de la circulation | chute d'arbre, arbre tombé | inaccessibles suite à la crue... | déposer, se débarrasser | départ sans un objet déposé... | invitation, rave partie, teuf... |
| Temps | confirmation que c'est actuel | confirmation que c'est actuel | confirmation que c'est actuel | confirmation que c'est actuel | confirmation que c'est actuel | confirmation que c'est actuel | événement à venir (prédiction) |
| Objet | qu'est-ce qui brûle | véhicule(s) | | rivière, canalisation... | | colis... suspect délaissé ... | |
| Lieu | lieu de l'incendie | lieu de l'accident | lieu de la chute | lieu de l'inondation | lieu du dépôt | lieu du colis suspect | lieu de la fête |
| Agent | | | | | | | organisateur |
| Cause | | | | pluies... | | | |
| Conséquence | | | | lieu ou voie fermé | Présence d'ordure... | | |
| Type | | | | | | | teuf rave partie |
| Date | | | | | | | absolue ou relative [23] |

TABLEAU 5.3 – Propriétés des événements à étudier

## 5.6 Conclusion

Nous avons abordé dans ce chapitre la notion d'événement et fait un tour de l'existant du point de vue des ontologies événementielles qui auraient pu être une réponse à notre problématique de représentation d'un événement comme une action située dans un lieu le plus précis possible et dans une temporalité acceptable pour déclencher des alertes pertinentes au plus tôt. Nous avons donc dû étendre un certain nombre de concepts et de relations afin d'obtenir des événements géo-chrono-localisés contextualisés par les tweets.

Cependant, il reste encore une optimisation à faire par rapport à cette représentation pour que la fusion des tweets non seulement infère de la connaissance de plus en plus complète sur les événements mais aussi pour restreindre la zone géographique pour son positionnement sur une carte lors du déclenchement de l'alerte correspondante.

# Chapitre 6

# Amélioration de la localisation des événements

## Sommaire


6.1 Nouvelles règles linguistiques pour GEOLsig . . . . . 149

6.2 Bases de données géographique . . . . . 158

6.3 Première expérimentation . . . . . 179

6.4 Conclusion . . . . . 181


Nous avons défini précédemment que l'événement est un triplet $A = action, L = Lieu, D = Date$. Nous avons aussi souligné le fait qu'il était primordial d'extraire chaque élément composant un événement le plus précisément possible pour générer une alerte adéquate.

Malgré la représentation ontologique dont nous disposons, il reste encore une part d'incertitude sur la véracité des lieux. Dans le but d'obtenir une géolocalisation plus précise de l'événement, nous avons eu recours à une base de données géographique (BDG) permettant de renvoyer les géolocalisations de chaque lieu détecté.

Dans un contexte de flux de tweets, cette base de données doit être accessible très rapidement malgré la complexité potentielle des requêtes. Il faut donc indexer cette BDG en préservant son aspect géographique.

Pour ce faire, nous avons mis en place le module *GEOLsig* dont l'objectif principal est

de compléter la détection des lieux suite au traitement fait par le module GAL qui reste incomplet malgré l'utilisation de dictionnaires et de l'ontologie.

Rappelons que, dans l'ontologie de GEOLSemantics, le lieu comprend les propriétés suivantes :

- le nom du lieu, exemples : Paris, île de France, France, etc. ;
- le type du lieu, exemples : ville, région, pays, etc. ;

Rappelons aussi que pour repérer les entités nommées de type lieu, nous utilisons deux méthodes différentes au niveau du GEOLsig :

- consultation de dictionnaires : pour les types de lieux, faire un thésaurus afin de regrouper les types sous des termes génériques (par exemple, lieu de culte pour église, basilique, chapelle, cathédrale, abbatiale, etc.), en reprenant ainsi ce qui a été réalisé au niveau du GAL à savoir :
  - détecter les lieux les plus connus (continents, océans, mers, pays, régions, capitales, grandes villes, ...) ;
  - typer les mots qui peuvent représenter un lieu comme église, basilique.
- patrons morphosyntaxiques :
  - par la typologie : 1ère lettre en majuscules,
  - par la morphologie : annonceur, nom propre,
  - par la syntaxe : après les prépositions (à, sur, etc.), avec certains verbes

Donc, le fichier de sortie après traitement par le module GAL contient déjà des lieux mais qui ne sont pas vérifiés. De plus, certaines annotations peuvent aussi être des lieux, comme les mots inconnus, les noms propres et les noms d'organisations. En l'absence d'un système de vérification, on peut perdre beaucoup d'informations sur les lieux, ou, *a contrario*, ajouter des lieux qui ne le sont pas.

Nous avons remarqué que les règles appliquées par GAL sont trop restrictives. Beaucoup de noms de lieux sont annotés en tant que mots inconnus ou affectés à d'autres catégories. Dans le contexte de cette étude sur les événements provenant de tweets, nous pouvons avoir

plus de permissivité sur les lieux potentiels, quitte à faire une vérification supplémentaire sur la pertinence des lieux trouvés. Pour ce faire, nous avons établi un certain nombre de règles linguistiques qui sont appliquées l'une après l'autre juste en sortie du GAL afin d'élargir la liste des lieux potentiels au-delà de ceux annotés en tant que tels par le GAL.

## 6.1 Nouvelles règles linguistiques pour GEOLsig

Afin d'établir les règles linguistiques, une procédure a été établie dans un premier temps sur la base d'un corpus d'environ 52 tweets localisés à Nice, et testés sur un corpus de plus de 18 000 tweets, toujours à Nice.

Les éléments de la sortie du GAL utilisés par les règles linguistiques sont :

- $Article = « le » | « la » | « les » | « l' »
- $Nom =
  - « +nom » : substantif ;
  - « +annp » : nom qui annonce qu'un nom propre suit ;
  - « +nomattr » : nom attribut ;
  - « +annppos » : nom qui annonce qu'un nom propre précède ;
  - « +an2np » : nom qui est suivi de deux noms propres, par exemple *match* ;
- EN complexe : ensemble de mots constituant une entité nommée, par exemple *tour Eiffel* ;
- EN simple : mot simple constituant une entité nommée, par exemple *Nanterre* ;
- Wsite : chaîne de caractères désignant une adresse de site web ;
- strType : nom désignant un type de voie dans une adresse, par exemple *rue* ou *avenue* ;
- lemma : normalisation de la forme de surface, par la transformation vers le masculin singulier pour les noms, et l'infinitif pour les verbes.

### 6.1.1 Vérification des entités nommées présentes

**Prise en compte d'un article**

**Algorithme 4 :** Algorithme de prise en compte d'un article

**Data :** *GN*, un groupe nominal

**Result :** *GN*, un groupe nominal

```
1 begin
2     if $Article (vérifier sur la forme de surface sans majuscule) devant ENloc,
       ENpers ou ENunk) ET (ENcomplexe | (ENsimple et pas +wsite dans les
       propriétés) then
3         tester un matching exact ;
4         if matching ne fonctionne pas then
5             tester GN sans l'article et en matching standard
6         else
7             if GN -$Article est ENloc ou ENpers ou ENunk then
8                 tester un matching standard
```

**Exemple 6.1.1.** Application de la première règle : si l'analyse a repéré l'*ENpers « Luc »* dans le tweet : *« ... entre Gonfaron et Le Luc, ... »*, il faudra chercher dans le SIG *« Le Luc »* en tant que forme exacte. Si celle-ci n'est pas trouvée, il faudra relancer une requête avec seulement *« Luc »* (cf. algorithme 4).

### 6.1.2 Vérification des autres groupes nominaux

N'appliquer les règles que s'il n'y a pas de recouvrement avec les précédentes chaînes testées (sans tenir compte des articles).

**Algorithme 5 :** Algorithme de choix du résultat EN loc

**Data :** $GN$, un groupe nominal

**Result :** $GN$, un groupe nominal

```
1 begin
2     if sortie / entree + nombre then
3         chercher sortie | entrée + nombre + NP
```

**Exemple 6.1.2.** Application de la règle 2 : EN loc : voie Mathis, puis GN la voie Mathis → ne pas retester (cf. algorithme 5).

**Algorithme 6 :** Algorithme de priorétisation des lieux renvoyés

**Data :** $GN$, un groupe nominal

**Result :** $GN$, un groupe nominal

```
1 begin
2     if « sortie » / « entrée » + nombre ) then
3         chercher « sortie »| « entrée » + nombre + NP * "sortie" OU "entrée" +
           nombre * "sortie" OU "entrée " + NP
```

**Exemple 6.1.3.** Application de la règle 3 : si dans le tweet, on trouve « entrée 54 Nice Nord », on cherche alors dans le SIG « entrée 54 Nice Nord », « entrée 54 » puis « entrée Nice Nord » selon l'algorithme 6.

**Algorithme 7 :** Algorithme de complétion d'une adresse

**Data :** $GN$, un groupe nominal

**Result :** $GN$, un groupe nominal

```
1 begin
2     if mot contenant la propriété + strType ET il existe un mot avec une
       majuscule dans la suite then
3         chercher ce mot avec la chaîne qui suit, et l'éventuel nombre qui le précède
```

**Exemple 6.1.4.** Application de la règle 4 : « rue + strType » → chercher « 19 rue Saint

François de Paule » (selon l'algorithme 7)

**Algorithme 8 :** Algorithme d'ajout des noms avec majuscule

**Data :** *GN*, un groupe nominal

**Result :** *GN*, un groupe nominal

```
1 begin
2     if $Nom avec une propriété "+geo") ET (nom avec majuscule OU nom suivi
      d'un mot avec majuscule, hors mots outils) ET (pas d'article indéfini avant le
      $Nom) then
3         chercher le nom et la chaine qui le suit (s'arreter, ou pas, lorsqu'on rencontre
          une préposition de lieu)
```

**Exemple 6.1.5.** Application de la règle 5 : soit le tweet « ... au niveau du pont de la Manda suite à un accident. », « pont » est un nom avec une propriété « +geo ». Il est suivi du mot « Manda » qui commence par une majuscule. On interroge donc le SIG avec « pont de la Manda » et « pont de la Manda suite à un accident » (selon l'algorithme 8).

**Algorithme 9 :** Algorithme de regroupements des lieux composés

**Data :** *GN*, un groupe nominal

**Result :** *GN*, un groupe nominal

```
1 begin
2     if +np avec une propriété +geo then
3         chercher le NP
```

**Exemple 6.1.6.** Application de la règle 6 : dans le tweet « ... dans le sens Marseille - Nice Circulation interrompue dans le tunnel dans ce sens. », on interroge le SIG avec « Marseille-Nice » (selon l'algorithme 9).

**Algorithme 10 :** Algorithme de regroupement des lieux séparés par des prépositions

**Data :** *GN*, un groupe nominal

**Result :** *GN*, un groupe nominal

```
1 begin
2     if préposition de lieu suivie d'au moins un nom en majuscule then
3         chercher la chaîne qui suit la préposition, et vérifier que le matching n'est
          pas seulement effectué sur les mots vides
```

**Exemple 6.1.7.** Application de la règle 7 : soit le tweet « ... est allumé sur l'autel de l'église Saint-Roch. », on interroge alors avec ce qui suit la préposition « sur », c'est-à-dire : « l'autel de l'église Saint-Roch », et pour éviter le problème des apostrophes, on ajoute un espace après les apostrophes : « l' autel de l' église Saint-Roch » (selon l'algorithme 10).

**Algorithme 11 :** Algorithme d'extraction des lieux inondés

**Data :** $GN$, un groupe nominal

**Result :** $GN$, un groupe nominal

```
1 begin
2     if verbe « déborder » a un GN sujet contenant au moins une majuscule then
3         chercher le GN sujet (chaîne qui précède le verbe)
```

**Exemple 6.1.8.** Application de la règle 8 : « Le petit Magnan déborde », on lance une requête avec « Le petit Magnan » (selon l'algorithme 11).

**Algorithme 12 :** Algorithme de regroupement des noms propres et des articles les précédant

**Data :** $GN$, un groupe nominal

**Result :** $GN$, un groupe nominal

```
1 begin
2     if catégorie +prenom précédée d'un $article (dans sa forme de surface) then
3         chercher la chaîne art+prenom
```

**Exemple 6.1.9.** Application de la règle 9 : « Le Luc » (selon l'algorithme 12).

**Algorithme 13 :** Algorithme de traitement des annonceurs de lieux

**Data :** $GN$, un groupe nominal

**Result :** $GN$, un groupe nominal

```
1 begin
2     if annonceur de lieu suivi d'un group nominal then
3         renvoyer le tout
```

**Exemple 6.1.10.** Application de la règle 10 : « Vers Le Luc » (selon l'algorithme 13).

**Algorithme 14 :** Algorithme de traitement des articles

**Data :** $GN$, un groupe nominal

**Result :** $GN$, un groupe nominal

```
1 begin
2     if au, à suivi d'un group nominal qui commence par une majuscule then
3         renvoyer le tout
```

**Exemple 6.1.11.** Application de la règle 11 : « je suis au chatelet » (selon l'algorithme 14).

### 6.1.3 Expérimentations et premiers résultats

Afin de tester la pertinence du nouveau système, nous avons effectué l'expérience suivante :

- récupérer des lieux qui existent sur OSM dans différentes catégories (867 lieux) ;
- lancer la récupération des tweets sur tous les lieux (maximum 100 tweets pour chaque lieu) ;
- enlever les doublons (même tweet mais deux comptes différents) (18 123 tweets) ;
- lancer les analyses de GEOLSemantics sans les règles linguistiques ;
- lancer les analyses de GEOLSemantics avec les règles linguistiques ;
- comparer les deux résultats.

Pour analyser les résultats obtenus, nous avons utilisé les trois indicateurs habituels que sont la précision, le rappel et la f-mesure.

**Précision** : c'est le rapport entre les vrais positifs et la totalité des positifs qu'ils soient vrais ou faux, soit $vp/(vp + fp)$. Ce taux indique la qualité des résultats.

**Rappel** : c'est le rapport entre les vrais positifs et la somme des vrais positifs et des faux négatifs, soit $vp/(vp + fn)$. Ce taux indique la quantité des résultats pertinents.

**F-mesure** : c'est le rapport entre la précision et le rappel selon $(precision * rappel)/(precision + rappel)$, dite moyenne harmonique. Elle indique la capacité à donner tous les résultats pertinents et à refuser tous les autres.

| | Tweets avec des lieux (extraction du lieu) | | Tweets sans lieux | |
|---|---|---|---|---|
| Annotation réel | 9820 | | 8303 | |
| Ancien système | 2103 | | 16020 | |
| | vp=1136 | fp=967 | vn=6536 | fn=9484 |
| Ajout de GEOLsig | 12306 | | 5817 | |
| | vp=7602 | fp=4704 | vn=3710 | fn=2107 |
| *Légende : vp : vrai positif, fp : faux positif, vn : vrai négatif, fn : faux négatif.* | | | | |

TABLEAU 6.1 – Résultats de comparaison pour l'extraction des lieux avec et sans GEOLsig

| | rappel | précision | f-mesure |
|---|---|---|---|
| Ancien système | 10,70% | 54,02% | 17,86% |
| Ajout de GEOLsig | 78,30% | 61,77% | 69,06% |

TABLEAU 6.2 – Statistiques sur les résultats de comparaison pour l'extraction des lieux avec et sans GEOLsig

Avec l'ancien système de GEOLSemantics (sans l'intégration de GEOLsig), seulement 17,86% (voir la f-mesure dans le tableau 6.2) des vrais lieux ont été annotés en tant que lieux avec un rappel de 10,70%. Ceci nous indique que de nombreux lieux ont été ignorés ; seulement 1 136 ont été détectés alors qu'il aurait dû en détecter 9 820 (cf. tableau 6.1) ont été correctement trouvés). La précision est de 54,02% ce qui indique que, même en essayant d'avoir des règles très strictes, le résultat peut nous induire en erreur puisque 967 lieux ont été détectés alors qu'ils ne le sont pas réellement. La confusion vient du fait que la plupart de ces lieux étaient des noms écrits en majuscules sans contexte. Ce type d'erreur est d'autant plus importante dans un contexte de tweets.

Nous avons eu une augmentation jusqu'à 69,06% (tableau 6.2, f-mesure) après les améliorations apportées par les règles linguistiques de GEOLsig. Ceci nous a encouragé à nous concentrer sur un système de règles linguistiques permettant d'extraire plus de lieux qui seront ensuite validés ou non par une base de donnée des lieux. Grâce à cela, l'ancien système est amélioré sur la détection des lieux, nous évitant une refonte totale de ce dernier.

Cette expérience concluante a constitué la première étape de la création du module *POSTgal* qui va être détaillé dans le chapitre 7.3.3. Cependant, il nous reste une notion à prendre en considération et pas des moindres dans le cas des réseaux sociaux à savoir le cas des

fausses informations (ou *"fake news"*) qui peuvent fausser l'analyse et qui pourraient malencontreusement générer de fausses alertes.

### 6.1.4 Cas des *Fake news*

Nous ne pouvons pas ne pas parler des fausses nouvelles (*fake news*) dans le contexte des réseaux sociaux. En effet, ce fléau gangrène de plus en plus les réseaux sociaux et a un impact important sur la société d'aujourd'hui. Le scandale de Cambridge Analytica, entre autres, fait partie des exemples les plus parlants. En effet, cette société a permis de récolter des informations sur des personnes en leur proposant de répondre à des questionnaires d'analyse de personnalités. Les groupes d'américains jugés indécis ont été bombardés plus tard de messages étant des *fake news* les poussant à voter pour Donald Trump [Seadle, 2019].

Lors de l'hackathon « Fake ou pas Fake ? » à la conférence CORIA-TALN [1] en 2018, nous avons, avec une équipe pluridisciplinaire, essayé de concevoir et déployer un système de détection de ces *fake news*. Nous avons effectué notre évaluation sur le jeu de données fournies par la société Storyzy au format .tsv. Storyzy [2] se spécialise dans la protection de l'image de marque par la détection de contenus suspects sur les sites Web qui peuvent potentiellement nuire à ces marques.

Le corpus contient des textes de divers sites Web en anglais (3 828 textes servent à la phase d'apprentissage et 1 277 pour réaliser les tests proprement dits) et en français (705 pour l'apprentissage, 236 pour les tests), ainsi que des transcriptions automatiques de vidéos Youtube en français (234 pour l'apprentissage, 78 pour les tests) sur le thème de la vaccination.

La tâche consistait à classer le contenu textuel en trois classes possibles : information correcte, information incorrecte, ou satire [Guibon *et al.*, 2019].

Afin d'obtenir un classifieur capable de mieux généraliser ses prédictions, nous avons essayé différentes représentations de données à savoir :

1. `https://er-tim.fr/en/node/3818`
2. `https://storyzy.com`

**Text ressemblance** (dt+tr) : le système de classification proposé était un système basé sur du *machine learning* en utilisant *scikit-learn*[3]. Plusieurs modèles ont été utilisés. Notre contribution a été d'ajouter le facteur linguistique à la classification. Pour ce faire, on a intégré au modèle des *features* sur les éléments linguistiques (sujet, verbe, adjectif, etc.) présents dans chacun des titres des articles et l'enchaînement de ces éléments. Nous avons remarqué que les *fake news* utilisent plus d'adjectifs et de superlatifs que les autres.

**Hashing Trick** (hv) : de plus, nous avons utilisé [Weinberger *et al.*, 2009] pour obtenir une autre représentation des données, en transformant des *features* arbitraires en indices dans un vecteur.

**FastText** (ft) : nous avons formé le modèle FastText [Bojanowski *et al.*, 2017] sur l'ensemble d'entraînement dans l'ordre pour éviter d'utiliser des données externes telles que des plongements déjà entraînés.

**Word2Vec** (wv) : nous avons également formé un modèle *skipgramme* de Word2Vec [Mikolov *et al.*, 2013a] avec un lot de 32 mots, 20 itérations et 300 vecteurs de dimension.

Nous avons remarqué qu'en utilisant une optimisation LGBM, (ft+wv+hv) obtient 91,39% de classification correcte tandis que (ft+wv+dt+tr) obtient 92,01%.

Après cet hackaton, nous avons compris qu'une approche linguistique va nous permettre de filtrer les *fake news*. Approfondir ce travail reste une tâche difficile et longue que nous n'aborderons pas plus en avant lors de ce mémoire. Cela reste une perspective pour des travaux futurs. Nous allons donc nous concentrer sur le perfectionnement de la géolocalisation des lieux maintenant.

3. https ://scikit-learn.org/stable/index.html

## 6.2 Bases de données géographique

### 6.2.1 Le système d'information géographiques

Un système d'information géographique (SIG) a pour spécificité de contenir des données spatiales et géographiques ; ces données ayant un format particulier. Les applications SIG permettent à l'utilisateur de requêter, de stocker et de modifier des données spatiales (géolocalisation d'un lieu) et non spatiales (type du lieu géolocalisé), d'analyser la sortie d'informations spatiales et de partager visuellement les résultats de ces opérations en les présentant sous forme de cartes. Les SIGs sont appliqués dans plusieurs domaines tels que la gestion logistique, les assurances, le marketing, les télécommunications, la télédétection, etc. [Maliene *et al.*, 2011].

Les emplacements qui se trouvent partout dans le monde peuvent être enregistrés en espace où les coordonnées spatiales sont $x$ (longitude), $y$ (latitude) et $z$ (élévation) ; et en temps si l'on se réfère à la toponymie. Représenter un lieu réel dans l'espace et le temps est une caractéristique clé du SIG qui a commencé à ouvrir de nouvelles voies de recherche et d'études scientifiques.

Les données d'un SIG peuvent être en accès libre (ou *open data*) ce qui est idéal puisque le coût induit par la cartographie de la terre entière serait considérable. La cartographie collaborative, également appelée cartographie participative des informations géographiques de volontaires (ou *"Volunteered Geographic Information (VGI)"*), est du contenu géo-spatial généré par des non professionnels à l'aide de systèmes de cartographie disponibles sur internet. Il s'agit d'un sous-ensemble de la néo-géographie. L'idée est qu'un groupe de personnes peut travailler ensemble pour recueillir des informations. Ces informations peuvent être très spécifiques, comme une carte de tous les cordonniers d'une région, ou plus génériques, comme une « carte gratuite du monde entier ». Il existe de nombreux outils pour créer des cartes collaboratives tels que WikiMapia, Ushahidi, Google Map Maker, Google MyMaps openStreetMap, etc.

### 6.2.2 Ressources de la base de données de GEOLSemantics

Une base de données de noms de lieux connus tels que des villes ou des villages avec leurs coordonnées géographiques est importante pour la plupart des approches de recherche d'informations géographiques et des services basés sur la localisation.

Plusieurs étapes ont été nécessaires pour créer la base de données géographique de GEOLSemantics. Dans ce qui suit, nous allons expliquer chaque étape à partir de la récupération des données libres de droit jusqu'à l'indexation et l'interrogation.

GEOLSemantics utilise des données collaboratives comme les données d'*Openstreetmap*[4].

Les données géographiques sont ensuite filtrées, structurées et normalisées afin de pouvoir combiner différentes sources. Ces étapes aussi seront discutées dans ce qui va suivre.

#### Constitution des ressources

**OpenStreetMap** Openstreetmap (OSM) est un projet collaboratif soutenu par la fondation Openstreetmap, organisation à but non lucratif enregistrée en Angleterre et au Pays de Galles, créée par Steve Coast en 2004, afin de créer une carte du monde modifiable et gratuite [Ramm *et al.*, 2011].

Les géo-données sous-jacentes à la carte sont considérées comme le résultat principal du projet. La création et le développement d'OSM ont été motivées par des restrictions d'utilisation ou de disponibilité des données cartographiques dans une grande partie du monde et par l'avènement de dispositifs de navigation par satellites portables bon marché.

**Contribution** : Il existe plusieurs sortes de contribution possible sur Openstreetmap : contribuer aux données cartographiques, développer des logiciels, signaler les problèmes, gérer le wiki opensteetmap, aider les utilisateurs et les développeurs, traduire les données. Les contributions doivent être :

— véritables : signifie qu'on ne peut pas contribuer avec des données inventées ;

4. openstreetmap.org

— légales : signifie qu'on ne peut pas copier les données protégées par le droit d'auteur sans autorisation ;

— vérifiables : signifie que d'autres peuvent s'y rendre et voir par eux-mêmes si les données sont correctes ;

— pertinentes : signifie qu'il faut utiliser des balises qui expliquent clairement aux autres comment réutiliser les données.

Openstreetmap impose peu de restrictions sur l'utilisation des balises. Il existe des normes de balisage mais qui évoluent au lieu d'être contraignantes. Openstreetmap encourage les interactions entre les contributeurs. Il valorise fortement les connaissances des personnes locales mais les modifications faites par des personnes éloignées du lieu considéré sont autorisées. Aucune autorisation n'est nécessaire pour effectuer un ajout à la base de données, ni pour modifier les contributions d'autres personnes. Il existe un système pour gérer les conflits entre les contributeurs, puisqu'Openstreetmap valorise la cohésion de la communauté plutôt que la perfection des données.

Jusqu'en juillet 2020, Openstreetmap comptait 6 627 391 d'utilisateurs et 7 989 263 755 points GPS.

**Format d'OSM** : on peut télécharger toutes les données de la planète [5] ou bien télécharger des données par pays, région, ville, etc. (par exemple, on peut passer par l'API Java Osmose [6] fournie par OSM ou par un site tiers comme GeoFabrick [7]). ON peut aussi choisir sur l'interface Openstreetmap une zone très réduite en passant par une *bounding box*.

Le fichier téléchargé est au format XML (6.1) selon le format suivant :

— une balise <osm> contenant la version de l'API "version" et le générateur qui a distillé ce fichier "generator" ;

— un bloc de balises "node" contenant les coordonnées des lieux en forme de point selon le système de référence WGS84 ayant des sous-balises portant des informations complémentaires sur le lieu comme le nom "name" (par exemple, entrée d'un immeuble.

5. wiki.openstreetmap.org/wiki/Planet.osm
6. Osmose
7. download.geofabrik.de

— un bloc de balises "way" contenant les coordonnées des lieux en forme de polygones ou multipoints, contenant des sous-balises comme le nom "name". Un "way" est un ensemble de "nodes" et chaque point qui compose le "way" est référencé dans les "nodes" (par exemple, tous les points composant un immeuble) ;

— un bloc de balises "relations" contenant les coordonnées des lieux en forme de polygones ou multipoints. Une "relation" contient des sous-balises comme le nom "name". Une "relation" est un ensemble de "nodes" et/ou de "ways" et/ou d'autres relations où chaque point qui compose la "relation" est référencé dans les "nodes" (par exemple, tous les immeubles composant un quartier).

Par exemple, une relation qui représente "île de France" = l'ensemble des ways qui composent les frontières d'île de France + l'ensemble des relations qui composent chacune les régions d'île de France + les points représentant le centre administratif et le centre géographique de l'île de France.

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6" generator="CGImap 0.0.2">
 <bounds minlat="54.0889580" minlon="12.2487570" maxlat="54.0913900" maxlon="12.2524800"/>
 <node id="298884269" lat="54.0901746" lon="12.2482632" user="SvenHRO" uid="46882" visible="true" version="1" changeset="676636" timestamp="2008-09-21T21:37:45Z"/>
 <node id="261728686" lat="54.0906309" lon="12.2441924" user="PikoWinter" uid="36744" visible="true" version="1" changeset="323878" timestamp="2008-05-03T13:39:23Z"/>
 <node id="1831881213" version="1" changeset="12370172" lat="54.0900666" lon="12.2539381" user="lafkor" uid="75625" visible="true" timestamp="2012-07-20T09:43:19Z">
  <tag k="name" v="Neu Broderstorf"/>
  <tag k="traffic_sign" v="city_limit"/>
 </node>
 ...
 <node id="298884272" lat="54.0901447" lon="12.2516513" user="SvenHRO" uid="46882" visible="true" version="1" changeset="676636" timestamp="2008-09-21T21:37:45Z"/>
 <way id="26659127" user="Masch" uid="55988" visible="true" version="5" changeset="4142606" timestamp="2010-03-16T11:47:08Z">
  <nd ref="292403538"/>
  <nd ref="298884289"/>
  ...
  <nd ref="261728686"/>
  <tag k="highway" v="unclassified"/>
  <tag k="name" v="Pastower Straße"/>
 </way>
 <relation id="56688" user="kmvar" uid="56190" visible="true" version="28" changeset="6947637" timestamp="2011-01-12T14:23:49Z">
  <member type="node" ref="294942404" role=""/>
  ...
  <member type="node" ref="364933006" role=""/>
  <member type="way" ref="4579143" role=""/>
  ...
  <member type="node" ref="249673494" role=""/>
  <tag k="name" v="Küstenbus Linie 123"/>
  <tag k="network" v="VVW"/>
  <tag k="operator" v="Regionalverkehr Küste"/>
  <tag k="ref" v="123"/>
  <tag k="route" v="bus"/>
  <tag k="type" v="route"/>
 </relation>
 ...
</osm>
```

FIGURE 6.1 – Format XML d'un fichier OSM

Dans la figure 6.2, nous pouvons voir une représentation d'une relation (île de France) qui peut être constituée de :

— un ensemble de "way" ; un way est représenté par un cercle vert, zoomé dans la figure 6.3 ;

— un ensemble de relations qui sont dans cet exemple les départements de l'île de

France où un exemple est mis en exergue avec le département des Yvelines qui est représenté en bleu ;

— un ensemble de "nodes" qui sont ici les centres géographique et administratif de l'île de France.

À noter qu'un "way" est composé d'une suite de "nodes".

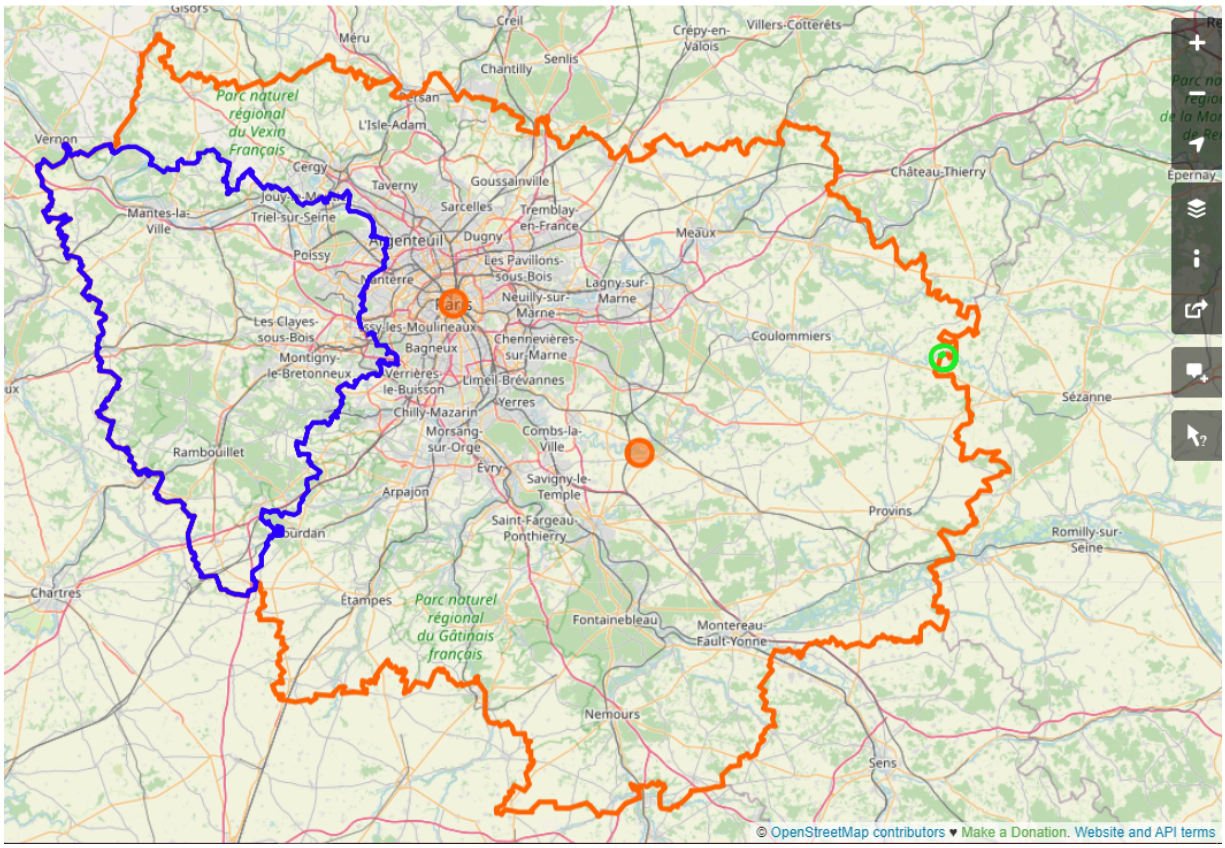



FIGURE 6.2 – Exemple d'une représentation way, node et relation dans OSM

**ci-dessous un schéma simplifié**

```
<BDDosm>
  <node id="1" lat="..." lon="..."/>
  <node id="2" lat="..." lon="..."/>
  <node id="3" lat="..." lon="..."/
  ...
  <way id="55">
     <nd ref="3">
     <nd ref="2">
  </way>
  ...
  <relation id="666">
     <member type="node" ref="1">
     <member type="way" ref="55">
  </relation>
  ...
</BDDosm>
```

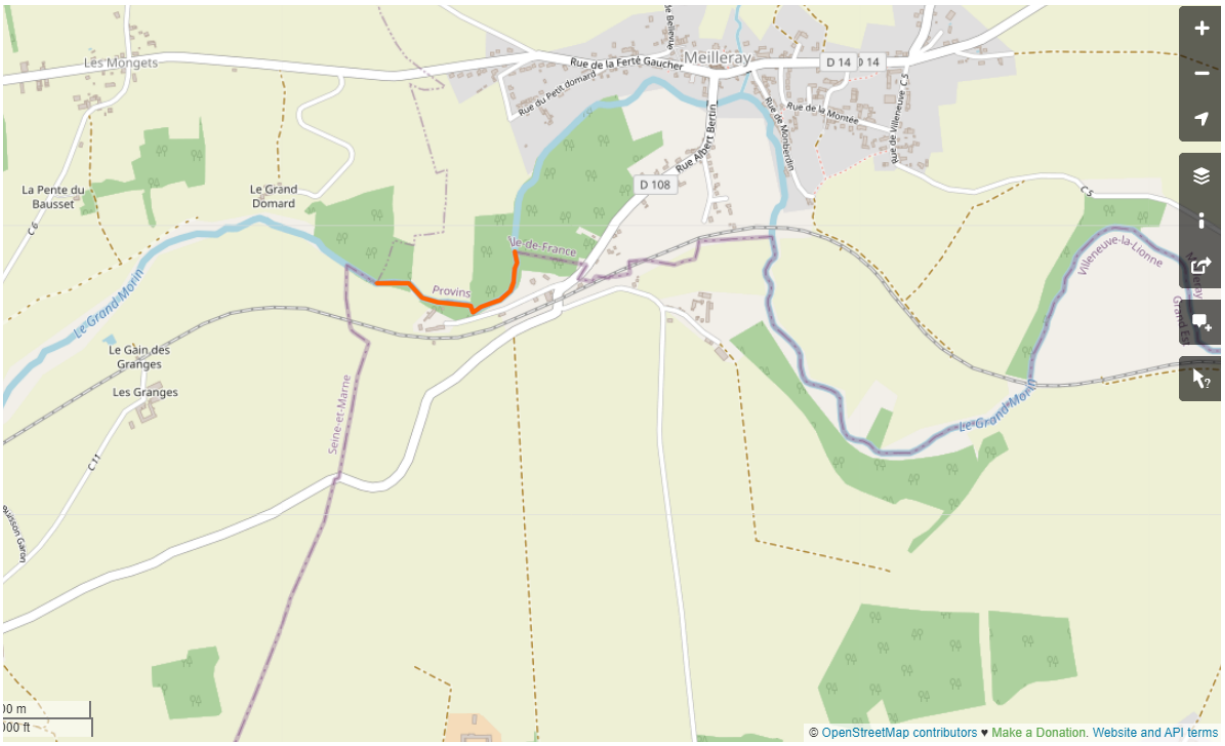



FIGURE 6.3 – Exemple d'une représentation way

Étant donné que les projets utilisés pour construire notre base de données sont des projets collaboratifs, grâce à la participation de volontaires, des erreurs sont à prévoir. Par exemple, il peut arriver qu'un même lieu soit déclaré deux fois par deux personnes différentes avec une proximité géographique minime. Il faut signaler aussi que dans ces projets libres, on trouve des types de lieux très détaillés comme un feu tricolore, un panneau signalétique, etc., des catégories que nous souhaitons ignorer pour notre base de données.

Afin d'éviter d'avoir une base de données trop volumineuse avec beaucoup de redondances, nous avons décidé de prendre toujours l'élément le plus englobant ; c'est-à-dire, si un lieu "way" est composé de "node", seul le way sera indexé et les nodes seront ignorés pour notre base de données.

De plus, afin d'éliminer (ou limiter) les redondances, un filtrage des lieux est effectué. Si deux lieux ont le même nom et la même catégorie, et qu'en appliquant une distance euclidienne, il s'avère qu'ils sont à moins de 3 mètres l'un de l'autre, les deux lieux seront regroupés avec une moyenne entre les deux distances. Nous avons choisi 3 mètres à titre expérimental mais cela risque de changer au fil des utilisations. En effet,

il faut souligner que la base de données OSM est réalisée par divers utilisateurs, nous avons remarqué, pour certains liens comme des arrêts de bus, qu'il pouvait exister deux entrées de deux personnes différentes. Nous avons donc décidé de regrouper ces entrées.

Dans l'exemple 6.2.2, il existe cinq lieux :

— trois lieux représentés par des points (1, 2, 3) ;

— un lieu en multipoints composé des points des nodes 3 et 2 (55) ;

— un lieu composé d'un point et d'un multipoint.

Afin de récupérer les coordonnées de chaque lieux, il faut :

— parcourir chaque node et stocker pour son *id*, sa longitude *lon* et sa latitude *lat* correspondantes ;

— parcourir chaque *nd* dans *way* et récupérer pour chacun sa longitude *lon* et sa latitude *lat* à partir des *node*, ($ref$ dans *way* est équivalent à la notion d'*id* dans *node*) ;

— parcourir chaque *member* de type *way* dans *relation* et récupérer pour chacun sa longitude *lon* et sa latitude *lat* à partir des way. Notons que $ref$ dans *relation* de type *node* est égal à *id* dans *node*.

**GeoNames** Les répertoires géographiques numériques (dits *gazetteers*) sont des dictionnaires géo-spatiaux de points géographiques avec des composants de base comme [Hill, 2000] :

— un nom : et toutes les variantes de ce nom de lieu ;

— un emplacement : coordonnées représentant un point, une ligne ou un emplacement géographique ;

— un type : selon les caractéristiques du lieu.

Selon Bol [2013], les gazetteers jouent un rôle important dans les systèmes pour

— l'accès géolocalisé à l'information et aux données ;

— l'identification des noms de lieux que nous utilisons dans le discours quotidien ;

— la levée des ambiguïtés : toponymes (noms propres désignant un lieu), résolution d'entités, géoparsing, classification, etc.

Un gazetteer peut, en outre, contenir une hiérarchie géographique, la population, des traductions du nom du lieu et d'autres connaissances géographiques sur un large éventail d'entités géographiques et géologiques. Le gazetteer gratuit le plus populaire est GeoNames, qui est utilisé dans un grand nombre de projets de recherche et de systèmes commerciaux [Ahlers, 2013] ;

GeoNames est une base de données géographique disponible et accessible via divers services Web. Le projet a été fondé par Mark Wick fin 2005 [Bol, 2013] sous une licence *creative commons*[8] ;

La base de données contient également des variantes orthographiques, ainsi que des orthographes non romaines de plusieurs de ces noms. Toutes les entités géographiques de la base de données contiennent des informations sur l'emplacement, la division administrative et la qualité. La base de données peut être utilisée à diverses fins, y compris l'établissement de l'orthographe officielle des noms de lieux étrangers, la cartographie, les systèmes d'informations géographiques (SIG), l'intelligence géo spatiale (GEOINT) et la recherche de lieux.

Les lieux sur GeoNames ne sont pas forcément sur Openstreetmap. GeoNames apporte une aide sur les lieux quand la requête est simplement un nom de lieu (lac, montagne, rivière, etc.) mais il est peu susceptible de localiser des rues individuelles[9]. De ce fait, dans notre étude, GeoNames est utilisé dans la création d'un module d'enrichissement permettant d'interagir avec le *Linked Open Data* (LOD). En effet, une fois les bonnes URIs identifiées, l'extraction est complétée avec des connaissances issues de DBpedia, Wikidata et GeoNames ;

Jusqu'en juillet 2020, GeoNames comptait 11 809 910 noms de lieux.[10]

8. `https://creativecommons.org`
9. wiki.openstreetmap.org/wiki/Geonames
10. www.geonames.org/statistics/

#### Autres ressources

Pour couvrir tous les lieux de la région de Nice, en plus de ceux fournis par GeoNames qui nous a servi à la détection des lieux dans la phase GAL et OpenStreetMap qui nous a permis de valider les lieux et obtenir leur géolocalisation, nous avons aussi utilisé la base des adresses nationales et les données ouvertes gouvernementales [11] pour compléter la base de données OpenStreetMap, notamment avec la liste des écoles et universités, et la liste des restaurants.

### Moteur de recherche du SIG

La recherche d'information fait partie intégrante de toute application. Effectuer des recherches sur une masse de données importante peut être difficile lorsque la rapidité, les performances et la disponibilité sont des exigences essentielles. Il existe de nombreux moteurs de recherche en *open source* sachant gérer du texte intégral tels que :

— **Apache Solr** [12] fournit toutes les capacités de recherche de Lucene via des requêtes HTTP. Il existe depuis 2004, ce qui en fait un produit mature avec une large communauté d'utilisateurs. Il offre des fonctionnalités puissantes telles que la recherche distribuée en texte intégral, la mise en facettes, l'indexation en temps quasi réel, la haute disponibilité, les fonctionnalités NoSQL, les intégrations avec des outils *Big Data* tels qu'Hadoop et la possibilité de gérer des documents en texte riche tels que Word et PDF.

— **Manticore search** [13] date de 2017. Il a été développé à partir de sphinxsearch [14] créé en 2001. Manticore offre une faible latence et une recherche sur du texte intégral à haut débit et un filtrage de flux à volume élevé.

— **RediSearch** [15], conçu en 2011, qui est spécialisé dans l'interrogation en temps réel. Il offre des fonctionnalités de recherche et de filtrage tels que les requêtes géo-spatiales, la récupération uniquement des ID (au lieu de documents entiers) et la notation

11. data.gouv
12. `https://lucene.apache.org/solr/`
13. `https://manticoresearch.com`
14. `http://sphinxsearch.com/`
15. `https://redislabs.com/modules/redis-search`

de documents personnalisée. Les agrégations peuvent combiner des opérations de mappage, de filtrage et de réduction / regroupement dans des pipelines personnalisés qui s'exécutent sur des millions d'éléments en un instant.

— **Xapian** [16] est une boîte à outils hautement adaptable qui permet aux développeurs d'ajouter facilement des fonctions avancées d'indexation et de recherche à leurs propres applications. Il prend en charge plusieurs familles de modèles de pondération et prend également en charge un riche ensemble d'opérateurs de requête booléens. La première version a été réalisée en l'an 2000.

— **OpenSearchServer** [17], disponible depuis avril 2009, il propose une série d'analyseurs syntaxiques, un *crawler* pour bases de données, pages web et documents riches ainsi qu'une interface permettant le développement de la plupart des applications en quelques clics.

— **Elasticlunr** [18] est un moteur de recherche léger en Javascript sur du texte intégral qui date de 2017. Il est disponible pour les navigateurs et la recherche hors ligne. Elasticlunr.js est un peu comme Solr avec moins de fonctionnalités, mais beaucoup plus léger, et avec une configuration flexible.

— **Elasticsearch** [19] compose avec Logstash et Kibana ce qu'on appelle "la pile ELK", alias Elastic Stack. Il étend les puissantes fonctionnalités d'indexation et de recherche de Lucene à l'aide des API RESTful et archive la distribution des données sur plusieurs serveurs à l'aide du concept d'index et de fragments. Elasticsearch est entièrement basé sur JSON. Cet outil est relativement récent (2010), mais il a gagné en popularité en raison de ses cas d'utilisation riches en fonctionnalités. Il offre une requête DSL [20] puissante, une recherche géographique, et une recherche distribuée sur du texte intégral.

Chacun de ces moteurs de recherche offre des solutions qui nous intéressent comme une analyse syntaxique, une recherche géographique, une utilisation hors ligne, une rapidité

16. `https://xapian.org/`
17. `https://www.opensearchserver.com/`
18. `http://elasticlunr.com/`
19. `https://www.elastic.co/`
20. https ://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl.html

de recherche et d'indexation. Les deux qui sortent du lot, et ceci par le nombre de leurs communautés respectives, sont Apache Solr et Elasticsearch. Tous les deux sont construits sur Apache Lucene. De ce fait certaines fonctionnalités sont très similaires dans les deux moteurs. Cependant, ils diffèrent considérablement en matière de déploiement, d'évolutivité, de langage de requête et de nombreuses autres fonctionnalités comme le montre le tableau 6.3 [21].

| | Solr | Elasticsearch |
|---|---|---|
| Installation et configuration | Facile à installer et à utiliser | Facile à installer et à utiliser. Plusieurs paquetages sont disponibles pour différentes plates-formes. |
| Recherche et indexation | Idéal pour la recherche sur du texte et les applications d'entreprise proches de l'écosystème *Big Data* | Idéal à la fois comme moteur de recherche de texte et comme moteur d'analyse grâce à son puissant module d'agrégation |
| Communauté | écosystème très vaste | écosystème florissant |
| Documentation | incomplète, obsolète | Bien documenté |

TABLEAU 6.3 – Comparaison entre Solr et Elasticsearch

À l'instar de Netflix [22], Stack Overflow [23] et LinkedIn [24], nous avons choisi Elasticsearch pour sa documentation, la disponibilité des membres de sa communauté, sa rapidité et surtout sa capacité à faire de la recherche géographique.

**Indexation de la base de données** Nous avons vu dans la section 6.2.2 les sources de données alimentant notre base de données géographiques nous permettant d'unifier toutes les représentations existantes. Puis, nous avons présenté notre démarche nous amenant à choisir ElasticSearch (cf. sous-section 6.2.2) comme moteur de recherche pour l'indexation dans un premier temps puis l'interrogation de cette base de données.

21. `https://sematext.com/blog/solr-vs-elasticsearch-differences`
22. `https://www.slideshare.net/g9yuayon/elasticsearch-in-netflix`
23. `https://www.infoq.com/news/2015/06/scaling-stack-overflow/`
24. `https://www.slideshare.net/TinLe1/elk-atlinked-in`

Nous allons montrer dans ce qui suit les étapes d'indexation de notre base de données.

Nous avons commencé par développer un outil (appelé OSMtoES.py), afin de parcourir chaque base de données en utilisant des parseurs pour les ressources qu'elles contiennent ; par exemple, *pyosmium*[25] pour OpenStreetMap. Nous avons utilisé la même méthode pour toutes les ressources. Nous allons donc expliquer la procédure pour openStreetMap qui a ensuite été transposée aux autres ressources en prenant en considération les différences de structures.

Il existe plus de 100 catégories dans OSM[26] allant de la plus générale comme la notion d'immeuble *'Buildings'* à la plus détaillée comme celle de feu tricolore *'traffic lights'*. Nous avons choisi 66 catégories que nous avons jugées pertinentes comme *'amenity'*, *'building'* et *'highway'* afin de constituer notre base de données. Notre parseur commence par récupérer les données sur les catégories sélectionnées pour :

— la catégorie "_categorie" : une liste dans laquelle est stockée la catégorie sélectionnée traduite en français. En effet, un lieu peut avoir plusieurs catégories synonymes qui sont stockées dans cette liste ;

— le nom "_name" : c'est une liste regroupant tous les noms disponibles d'un lieu ("name", "old_name", "short_name", "official_name"...) ;

— les coordonnées de la localisation "coordinates" : c'est une liste des coordonnées (longitude, latitude) du lieu, combinée au type du lieu (point, cercle, multipoints, polygone) afin de former le paramètre "_location" ;

— le numéro d'adresse "_addrhousenumber" : String contenant le numéro de la rue si le numéro de la rue est présent le stocker ;

— l'adresse "_addrstreet" : chaîne de caractères contenant le nom de la rue si le nom de la rue est présent le stocker ;

— le code postal "_addrpostcode" : chaîne de caractères contenant le code postal si le code postal est présent le stocker ;

— la ville "_city" : chaîne de caractères contenant ville si l'adresse de la ville est présente la stocker ;

25. `https://docs.osmcode.org/pyosmium/latest/intro.html`
26. `https://wiki.openstreetmap.org/wiki/OpenStreetBrowser/Category_list`

— le pays "_country" : String contenant pays si le nom du pays est présent le stocker ;

— l'adresse complète "_address" : chaîne de caractères contenant adresse complète si l'adresse est présente la stocker sinon regrouper "_addrhousenumber" "_addrstreet" "_addrpostcode" et "_city" ;

— la valeur de la catégorie englobante par rapport à "_name" dans "_is_in" : liste d'éléments trouvés généralement dans geonames, qui pointe vers le lieu englobant ; par exemple, si "_name"=Paris alors "_is_in"=France.

Le code json du listing 7.5 montre un exemple d'un lieu qui sera indexé. Une fois tous les lieux transformés en json, il faut les regrouper. En effet, le même lieu peut être mentionné dans plusieurs ressources. Un autre programme développé par GEOLSemantics compare tous les json de tous les lieux.

Si certains lieux partagent les mêmes éléments "_name", "coordinates" et "_categorie", on ne garde qu'un seul json. Si il n'y a que "coordinates" qui est différent, la distance euclidienne est calculée entre les lieux. S'ils sont à moins de 20 mètres, les lieux sont combinés en calculant le centroïde [27] et le json correspondant est stocké. Dans cette phase, chaque document json obtient un identifiant (id) unique pour lui permettre d'être stocké sur ElasticSearch sans être écrasé par un autre json. Nous aurions pu faire la combinaison des ressources sur ElasticSearch et supprimer les doublons mais nous avons préféré gérer cette partie avec nos propres outils afin de pouvoir contrôler le processus, étudier les bugs potentiels et avoir une traçabilité des résultats. En conséquence, nous avons décidé que l'utilisation d'ElasticSearch ne devrait servir qu'à l'indexation et au requêtage de la base de données.

Avant d'indexer la base de données sur ElasticSearch, il faut créer un *mapping* de notre base de données. Le *mapping* est le processus de définition de la manière dont un document et les champs qu'il contient sont stockés et indexés. Par exemple, par un *mapping*, on peut :

— définir les types des champs indexés (string, entier, etc.) ;

— définir les champs qui contiennent des nombres, des dates ou des géolocalisations ;

— définir la langue des champs indexés.

27. La moyenne des points qui composent le lieu [moyenne longitude, moyenne latitude]

Pour notre base de données, nous avons défini dans notre *mapping* ce qui suit :

— l'index, c'est-à-dire le nom donné à l'ensemble des documents json, qui permet l'indexation et fait partie de la requête afin de récupérer un ou plusieurs documents json, comme on va le voir dans la requête 6.2.2 ;

— le type de chaque paramètre ;

— la langue de la requête (français) ;

— une liste de mots vides en français.

Une fois le mapping de la BD réalisé[28], on lance l'indexation des lieux[29]. Cette étape est plus au moins longue selon le nombre de lieux à indexer.

**indexation et recherche** La structure du format JSON permet d'indexer tout le contenu des documents afin de faciliter la recherche sur ce contenu plus tard. Ceci facilite la recherche qui peut se faire de façon ciblée sur des propriétés précises. Par voie de conséquence, les recherches sont réalisable de deux façons, soit sur n'importe quelle propriété, soit en cherchant du contenu sans indiquer les propriétés qui les contiennent. ElasticSearch traite chaque requête selon le principe suivant [JVCJuvénal, 2020] :

— la requête arrive dans le cluster ElasticSearch ;

— la requête est réceptionnée par le nœud maître du cluster :

— la requête est transférée à tous les nœuds qui possèdent une partition de l'index ;

— sur chacun de ces nœuds, une instance d'Apache Lucene qui y tourne effectue l'opération de recherche sur la partition ;

— les résultats correspondants sont renvoyés au nœud maître ;

— le nœud maître réceptionne les résultats de chaque nœud, les agrège et les retourne.

Ce procédé assure que la charge de calcul des requêtes est équitablement répartie entre les nœuds du cluster.

28. `https://www.elastic.co/guide/en/elasticsearch/reference/6.8/indices-put-mapping.html`
29. `https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-create-index.html`

Nous exécutons les requêtes via l'API REST d'ElasticSearch en lui transférant les requêtes HTTP des URI en cURL[30].

curl -XPOST *http* : *localhost* : 9200/index/_search -d'{corps de la requete}

**Type de requêtes exécutées** Nous effectuons deux types de requêtes sur notre base de données à savoir des requêtes directes et des requêtes géographiques. Les premières agissent sur un des paramètres "name" du json, ("_categorie", "_name", "_address") afin de récupérer les coordonnées comme dans le script du listing 6.1.

Listing 6.1 – Exemple d'une requête simple

```
{
   "script":{
      "lang":"mustache",
      "source":{
         "query":{
            "match":{
               "{{name}}":"{{value}}"
            }
         }
      }
   }
}
```

Nous faisons aussi des requêtes géographiques fournies par ElasticSearch afin de désambiguïser la présence de prépositions. Par exemple, « Je suis aux alentours de Paris » doit renvoyer une zone autour de Paris comme exprimé dans le listing 6.2.

Listing 6.2 – Exemple d'une requête géographique

```
GET /example/_search
{
  "query": {
```

30. https://curl.haxx.se/docs/history.html

```
    "bool": {
      "must": {
        "match_all": {}
      },
      "filter": {
        "geo_shape": {
          "location": {
            "shape": {
              "type": "envelope",
              "coordinates": [ [ 13.0, 53.0 ], [ 14.0, 52.0 ]
                  ]
            },
            "relation": "within"
          }
        }
      }
    }
}
```

ElasticSearch propose plusieurs formules de requêtes géographiques que l'on continue encore aujourd'hui à explorer [31].

**Sélection du résultat final** ElasticSearch offre une requête rigide sur les éléments de type chaîne de caractères avec lesquels il faut un appariement exact pour renvoyer le résultat. Cette méthode bien qu'efficace ne nous aide pas dans un contexte de tweets puisque certains lieux sont mal orthographiés. Nous avons donc opté pour une requête plus souple.

On avait déjà opté pour plus de souplesse pour les règles d'extraction des lieux puisque

31. https://www.elastic.co/guide/en/elasticsearch/reference

nous pouvions les valider avec notre base de données. Par exemple si un nom commence par une majuscule au milieu de la phrase, on teste s'il s'agit d'un lieu (cf. l'exemple concret ci-dessous).

**Exemple 6.2.1.** Soit l'extrait de tweet « *Accident de voiture Photo www.pic.com* », *Photo* sera extrait comme un lieu possible et va être utilisé comme un critère d'interrogation sur notre base de données.

Une requête avec ElasticSearch ne renvoie pas un seul résultat mais une liste de résultats possibles. Par exemple, en testant sur un corpus de 1000 tweets avec des lieux se trouvant à Nice, dans 100% des cas le bon résultat se trouvait dans les 10 premiers résultats renvoyés par ElasticSearch, mais pas toujours comme premier résultat. Nous avons donc développé un système de calcul de points par bonus/malus notant la pertinence du résultat. À chaque résultat parmi les 10 premières propositions d'ElasticSearch sera attribuée une note (positive ou négative) selon différents critères. Le résultat ayant le plus grand nombre de points sera renvoyé. Si tous les résultats obtiennent une note négative, la requête ne sera pas considérée comme concernant un lieu et sera exclue.

Les critères retenus sont :

- traitement selon les caractères :
  - poids positifs :
    - appariement complet avec et sans les mots vides —-> requête = GEOLSemantics | résultat = GEOLSemantics ;
  - poids négatifs :
    - distance de Levenshtein entre la requête et le résultat —-> requête = GEOLOSemantics | résultat = GEOLSemantics ;
    - distance de Levenshtein entre la requête et le résultat+sa catégorie —-> requête = entreprise GEOLSemantics | résultat = GEOLSemantics, catégorie = entreprise ;
- traitement selon les mots :
  - poids positifs :

- le résultat est contenu dans la requête avec et sans les mots vides —-> requête = entreprise (de) GEOLSemantics | résultat = GEOLSemantics ;
- la requête est contenue dans le résultat avec et sans les mots vides —-> requête = GEOL | résultat = GEOL SEMANTICS ;
- résultat sans ordre est contenu dans la requête sans ordre avec et sans les mots vides requête = GEOL SEMANTICS | résultat = SEMANTICS GEOL ;
- mot en commun est un nom propre ou prénom requête = SEMANTICS | résultat = GEOL SEMANTICS ;

— poids négatifs :

- le mot qui n'est pas en commun est un nom propre ou prénom —-> requête = Paris photographie | Parisian photographie ;

— traitement selon les catégories des résultats :

— poids positifs :

- la catégorie du résultat est dans la requête = PHOTOS |résultat = labo photo catégorie = photo ;
- la catégorie du résultat est dans la liste des catégories prioritaires :
  - les catégories sont classées par ordre de priorité ;
  - catPriority = "ville,commune,banlieue,quartier,village,avenue,boulevard,rue,voie piétonne,place,immeuble,théâtre" ;

— poids négatifs :

- la catégorie du résultat est dans la liste des catégories minoritaires :
  - catMinority = "arret,bus,tram,cafe" ;

— exemple sur les catégories prioritaires et minoritaires :

- requête : Je suis à République
- résultat ;
  - république catégorie bus
  - république catégorie place

— le résultat renvoyé : république catégorie place puisque place est dans la catégorie prioritaire.

### Résultat final

Le système GEOLsig permet de transformer du texte en géolocalisation en temps réel ou en différé en signalant l'action qui vient de se passer dans la géo-chrono-localisation signalée, tout en prenant en considération la sémantique des prépositions et la problématique de l'orthographe des tweets.

Nous avons et nous continuons à développer ce système pour les besoins du projet *Safecity*. Nous l'avons aussi appliqué dans d'autres contextes grâce à sa modularité et sa généricité en proposant d'autres extensions d'utilisation comme énoncé dans les paragraphes suivants.

**Envoi d'un json d'alerte aux partenaires du projet *Safecity*** Comme souligné précédemment, le projet *Safecity* regroupe plusieurs partenaires, chacun travaillant sur une problématique afin d'accumuler des éléments permettant de signaler et d'aider à résoudre une alerte. GEOLSemantics a pris en charge la tâche de détection des événements présents dans les tweets.

Pour ce faire, le module GEOLsig a été branché avec le programme de récupération des tweets par *bounding box* sur la ville de Nice, renvoyant un document json (listing 7.5) aux partenaires à chaque événement détecté nécessitant une alerte.

**Interface de traitement** Afin de faire une démonstration spécifique à GEOLSemantics sur un corpus de tweets collecté auparavant, une interface a été développée qui prend en argument une liste de tweets contenant des métadonnées sur la date du tweet afin d'afficher le lieu et l'action présents dans le tweet sur une carte *OpenStreetMap*. Elle permet à l'utilisateur de charger un fichier de tweets. Ensuite, elle affiche les tweets sur la droite de l'écran. L'utilisateur peut ensuite lancer l'analyse sur un tweet au choix. L'interface affiche

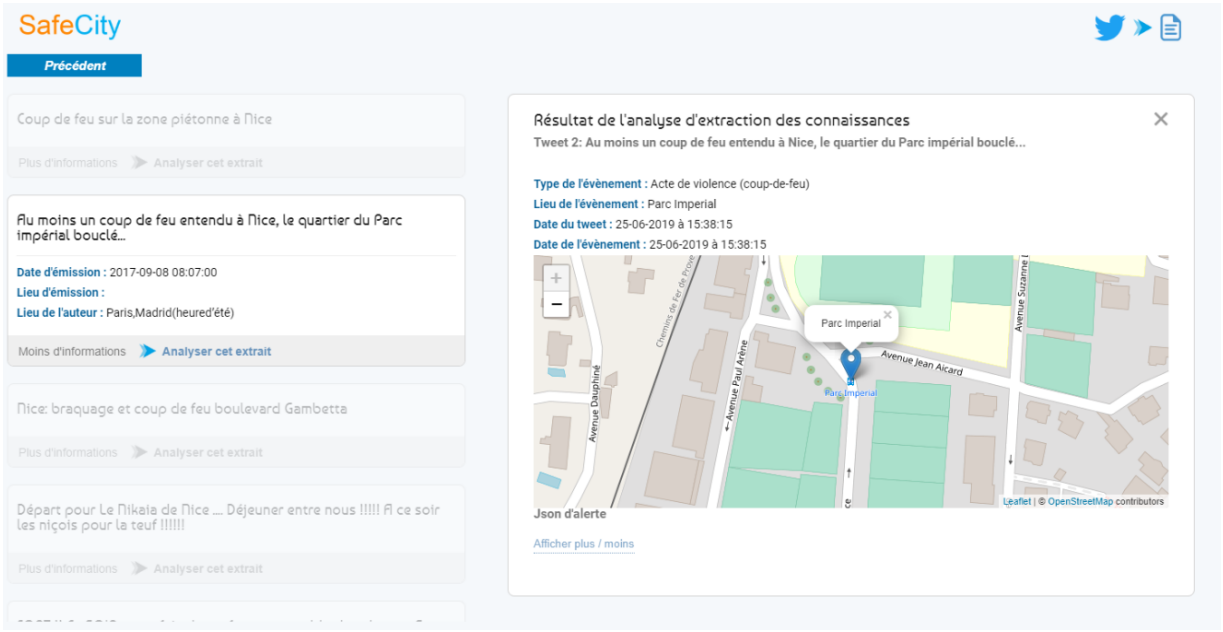



FIGURE 6.4 – Interface de géolocalisation des lieux

une carte avec une géolocalisation de l'événement s'il est présent dans le tweet (voir figure 6.4).

Par cette interface, une analyse et un regroupement d'information de plusieurs tweets (*cf.* regroupement des tweets 5.4.1) est faite. Cependant, elle ne permet pas de lancer les alertes proprement dites puisque les tweets ne sont pas récupérés en direct.

Cette interface nous a permis de mettre au point le module GEOLsig et d'affiner les résultats de géolocalisation renvoyés par ce dernier.

**Alertes twitterBOT** Afin de générer les alertes, étape suivante de l'affichage sur une carte, Une autre application a été conçue et développée en utilisant l'API twitter. Cette dernière, en plus de la collecte des tweets, permet de contrôler un compte Twitter automatiquement et d'effectuer les opérations de tweet, retweet, messagerie directe...

En utilisant l'API twitter, nous récupérons les tweets sur une *bounding box* de la France en temps réel. Puis, nous filtrons sur la langue française en utilisant le paramètre *"langue"* de l'API.

Nous analysons chaque tweet. Si un événement est détecté et géolocalisé, le tweet en

question est retweeté avec une légende qui est composée comme suit : *"événement à lieu"*. La figure 6.5 est une capture d'écran de l'outil geolBOT (qui est en mode privé pour le moment).




FIGURE 6.5 – Retweet automatique d'un événement par geolBOT

En utilisant les résultats de la recherche effectuée dans la section 4.5, nous pourrons aussi ajouter une mention à une autorité (police, SAMU, pompiers, ville, etc.) à chaque légende. À l'heure actuelle, cet outil fonctionne par l'intermédiaire d'un compte privé sur Twitter pour nous permettre d'améliorer le module GEOLsig. Une fois finalisé, il sera utile (après une adaptation des règles d'extraction) pour des journalistes ou des entreprises qui auront alors à leur disposition un outil pouvant retweeter des tweets intéressants en temps réel, en prenant en considération la sémantique du tweet afin d'améliorer leurs procédures de recherche actuelles fondées sur une recherche par mots-clés ou par hashtags. En ce qui concerne la recherche par mots-clés, elle n'est souvent pas pertinente comme nous l'avons mentionné dans la section 4.4.1. En ce qui concerne la recherche par hashtag, elle est souvent polluée. Beaucoup de personnes essaient de profiter de la popularité d'un hashtag afin de placer des tweets de publicité qui n'ont aucune relation avec le thème du hashtag.

La procédure d'écriture de règles d'extraction devrait aussi être optimisée en utilisant BERT afin de générer des règles automatiquement. Cela devrait rendre l'adaptation à un thème donné plus rapide.

## 6.3 Première expérimentation

Pour vérifier nos dire, nous allons procéder à une première expérimentation sur un échantillon de 1 495 tweets, analysé avec notre système et BERT. Rappelons que BERT [32] [Devlin *et al.*, 2018b] est un modèle contextuel de langues, fondé sur les réseaux de neurones, développé chez Google et diffusé en *open source* en 2018. Le succès révolutionnaire de BERT et de la classe de modèles appelés "transformeurs" (BERT, ALBERT, RoBERTa, GPT, GPT-2, etc.), a permis de mettre au point des modèles linguistiques capables d'atteindre des performances record dans les différentes tâches du NLU "Natural Language Understanding". CamemBERT et FlauBERT sont deux variantes qui offrent les meilleurs résultats pour la langue française. L'avantage d'utiliser ces modèles est la réduction du coût de préparation du corpus d'apprentissage car ils sont pré-entraînés sur des corpus très volumineux en français issus de différentes sources (Wikipedia, livres, internet, journaux, etc.). Notons que le plongement contextuel (*Contextual embeddings*), comme BERT, va au-delà des représentations des plongements de mots (*Word embedding*), comme Word2Vec, et atteint des performances révolutionnaires sur un large éventail de tâches de traitement du langage naturel puisqu'il attribue à chaque mot une représentation basée sur son contexte. Liu *et al.* [2020] ont examiné les modèles existants d'intégration contextuelle.

Pour la comparaison avec notre système, nous avons affiné le réseau de neurones fourni par CamemBERT en lui ajoutant une couche de sortie concernant la reconnaissance des deux entités "ACTION" et "LOCATION". Nous avons entraîné ce nouveau modèle sur un corpus d'apprentissage composé de 1195 tweets sur les inondations dans la région de la Côte d'Azur en France, sur une période allant de 2006 jusqu'à 2019. Le corpus intégral est composé de 1495 tweets, annotés en précisant le lieu, le déclencheur de l'action *inondation* (inondation, montée des eaux, etc.), et si le tweet représente un événement ou non. 300 tweets ont été isolés après annotation et utilisés comme corpus de test pour les deux systèmes.

Pour notre solution, nous avons pris un corpus de développement de 118 tweets (10%) issus du corpus d'apprentissage de BERT pour affiner nos règles. Après l'affinage des règles sur

32. Bidirectional Encoder Representations from Transformers

les deux systèmes à comparer, nous avons lancé les analyses sur le corpus de test restant de 300 tweets. Les résultats sont présentés dans le tableau 6.4.

Afin de valider les fusions d'événements, plusieurs des tweets utilisés présentaient des similarités partielles ou totales (considérant l'action et les localisations temporelles et géographiques). Ceci nous a permis de confirmer le bon fonctionnement de la fusion de tweets. Les résultats ont également pu être analysés afin de s'assurer que l'intégralité des champs JSON du regroupement - identifiant, dates, intersection - étaient bien remplis et au bon format.

### 6.3.1 Analyse des résultats

Dans un premier temps, nous avons remarqué que BERT est efficace pour détecter des lieux et des actions mais présente des lacunes en sémantique. La F-mesure de la détection des actions seules (resp. lieux) est de 88% contre 70% pour GEOLSemantics (resp. 96% contre 88%). En effet, un message peut inclure un déclencheur d'action et un lieu sans pour autant représenter un événement (négation, événement très ancien, métaphore, etc.), tandis que notre système a une plus grande précision (88%) et ne renvoie que les événements sémantiquement corrects. Par exemple, *"Mon balcon est inondé, c'est une piscine olympique le truc"* est renvoyé par BERT, mais pas par GEOLSemantics.

Puisque notre outil est plus correct dans son traitement des événements qui ne doivent pas déclencher d'alerte, nous avons pris les résultats de GEOLSemantics (GEOL) sur les non-événements et les résultats de BERT sur les événements, constituant ainsi un système hybride qui arrive à 89% de F-mesure tout en étant meilleur en rappel et en précision que chacun des systèmes exécutés individuellement. Enfin nous avons étudié les cas où au moins l'un des systèmes avait une bonne réponse. Cette approche a obtenu 95% de F-mesure. Il semble que les deux systèmes soient complémentaires et qu'il faille les configurer selon le contexte. Concernant les temps d'exécution, BERT est plus rapide, ce qui s'explique par la quantité de traitements que nous effectuons. Nous pallions ceci en effectuant de la parallélisation des traitements à tous les niveaux d'analyse, mais aussi en ajoutant une première étape qui applique un filtre « à grandes mailles » sur le flux des tweets d'entrée,

| | GEOL | Bert | Non événement–>GEOL<br>évenement–>BERT | Combinaison<br>GEOL/BERT |
|---|---|---|---|---|
| Rappel | 86% | 90% | 90% | 99% |
| Présision | 88% | 74% | 89% | 91% |
| F-mesure | 87% | 80% | 89% | 95% |

TABLEAU 6.4 – Tests de performance des systèmes GEOLSemantics et BERT

afin d'analyser que ceux qui ont une chance de contenir des informations pertinentes. Ce filtre s'appuie sur une liste de mots-clés pouvant représenter un événement à extraire.

## 6.4 Conclusion

Nous avons montré qu'un processus de traitement complet combinant des approches linguistique et sémantique afin de compléter les informations composant un événement par recoupement et regroupement est possible. Un système hybride nous permet d'obtenir des résultats très satisfaisants pour une première expérience, résultats améliorables en configurant mieux BERT et en rendant notre système encore plus robuste.

Diverses améliorations sont planifiées pour l'algorithme de regroupement d'événements ayant un contexte commun partiel. Par exemple, regrouper des événements de localisation géographique et de temporalité similaires mais dont l'action représentée est différente. En effet, malgré cette divergence d'action, les autres éléments détectés laissent suggérer que de tels événements sont liés d'une manière ou d'une autre. D'autre part, il faut affiner la modélisation des zones géographiques des événements à fusionner. En effet, dans plusieurs cas, utiliser l'intersection entre chacun des lieux déterminés s'avère moins pertinent que de conserver une zone trouvée dans un des événements à fusionner.

Nous avons expérimenté notre système sur un corpus d'inondation. Pour nos travaux futurs, nous souhaitons étudier la pertinence de notre système dans d'autres contextes. En effet, le tableau 6.4 a montré que d'autres contextes peuvent influencer le choix du système.

Dans la section 7.4 du chapitre 7 nous allons utiliser un corpus construit à partir de tous

les événements avec des incendies sur la Côte d'Azur pour une période allant de 2006 à 2020. Ceci va également nous permettre d'étudier les capacités de BERT pour la détection des multi-événements, tâche déjà développée dans notre système. Nous allons aussi tester la rapidité et la précision des deux systèmes en simulant un flux de tweets en temps réel comportant beaucoup de bruit (tweets non pertinents).

# Quatrième partie

# Interprétation de groupes de messages et suivi d'événements

# Chapitre 7

# Suivi des messages

## Sommaire


7.1 Traitement contextualisé des lieux . . . . . 185
7.2 Émission d'alerte . . . . . 188
7.3 Adaptation de la chaîne de traitement GEOLSemantics . . . 189
7.4 Expérimentations . . . . . 203
7.5 Conclusion . . . . . 210


Afin de garder la trace de l'évolution d'un événement, il est nécessaire de disposer de l'ensemble des messages qui le constituent. Ce suivi est important autant dans sa dimension spatiale que temporelle.

## 7.1 Traitement contextualisé des lieux

De façon générale, le traitement linguistique des lieux dans une phrase est difficile.

Dans le cas des tweets, même s'il est de nature très courte, il peut néanmoins transmettre des informations complexes surtout en ce qui concerne les lieux. Par exemple, on peut trouver :

— un lieu : *« Je suis à Paris. »* ;

— plusieurs lieux : *« Je suis à Paris, ma voiture est à Lyon. »*

— un lieu composé : *« Je suis entre Paris et Lyon. »* ;

— un lieu flou : *« Je suis aux environs de Paris. »*.

Il est donc primordial de mener une étude exhaustive concernant les lieux et les prépositions des lieux. Nous considérons que les notions de localisation et de lieu sont deux concepts différents. Dans cette étude, un lieu est le nom extrait du tweet, présent dans notre base de données des lieux. Sa localisation est le résultat de l'analyse des éléments qui le contextualisent, plus précisément les prépositions. Plusieurs lieux peuvent générer une seule localisation. Dans l'exemple : *Je suis à Paris, ma voiture est à Lyon*, on trouve deux localisations, Paris et Lyon. En revanche, dans l'exemple, *Je suis entre Paris et Lyon*, on a une seule localisation qui est la zone entre Paris et Lyon.

Cette localisation sera affichée sur la carte de façon différente selon son type (un lieu, plusieurs lieux, lieu composé, lieu flou, etc.). Certains de ces cas sont représentés sur la figure7.1.

L'utilisation de plusieurs lieux par le rédacteur du tweet peut être volontaire de sa part pour lever toute ambiguïté. Par exemple, quand il dit : *Je suis à Paris, France.*, il lève l'ambiguïté en ajoutant un second lieu plus général à savoir le pays. Dans le cas où l'on veut volontairement lever une ambiguïté, c'est que, par défaut, l'interprétation courante en France n'est pas la bonne. Par exemple, on parlera de *Paris / Texas* pour désigner la ville américaine. Il y a beaucoup de villes américaines qui portent des noms de villes européennes.

Malheureusement, ce n'est pas toujours le cas. Il peut rester beaucoup d'ambiguïtés pour différentes raisons : lieux portant le même nom mais dans des villes, des communes, des départements, des régions, des pays différents. Par exemple, *Montreuil* correspond à au moins deux villes dans des communes différentes. L'une se situe dans la commune de Montreuil (Montreuil-sous-bois) et dans le département de Seine-Saint Denis, l'autre dans la commune de Montreuil (cf. Montreuil-sur-mer) dans le département du Pas-de-Calais.

Autant pour l'émetteur que le(s) récepteur(s) du message, il peut ne pas y avoir d'ambiguïté car ils se connaissent ou peut-être qu'ils ont eu d'autres moyens d'échanger (télé-

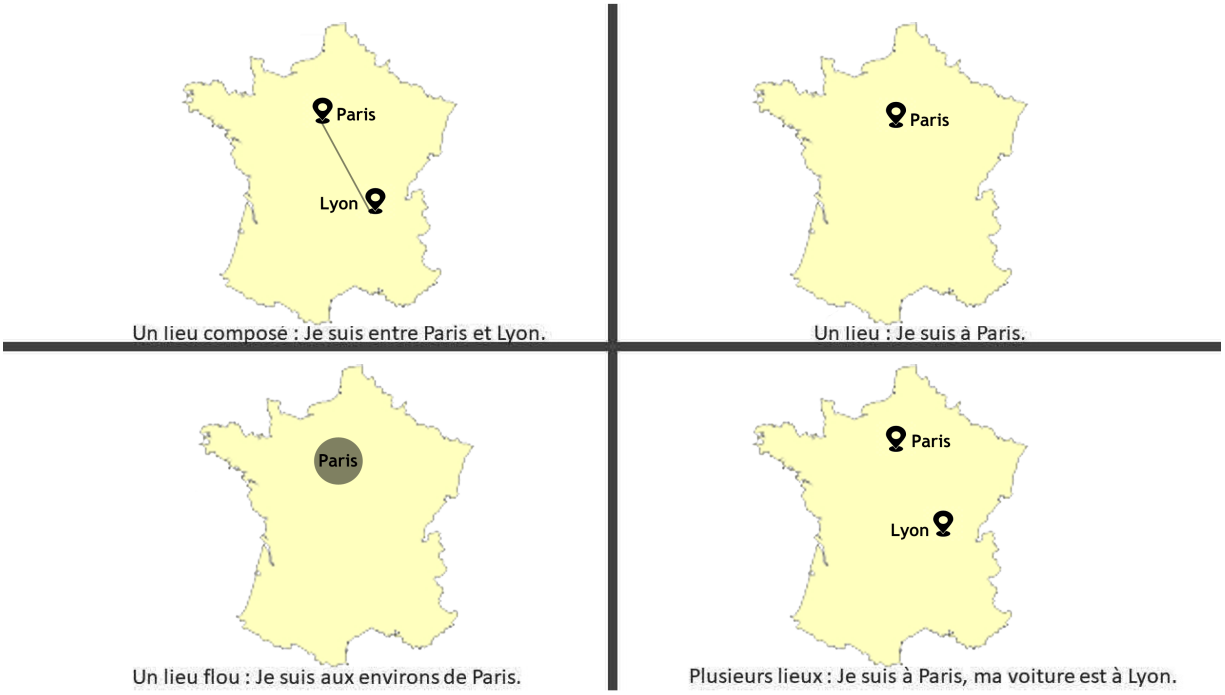



FIGURE 7.1 – Type de déclaration de lieux

phone, etc.) pour lever une ambiguïté éventuelle, autant pour notre traitement linguistique, nous devons faire des analyses plus poussées en l'absence de la connaissance du contexte pour définir le bon choix parmi les différents lieux avec le même nom. Pour cela, nous pouvons préciser une localisation par l'utilisation des métadonnées de l'émetteur quand elles sont disponibles ou nous appuyer sur le découpage géographique administratif. Pour ce faire, il est possible d'utiliser les relations d'inclusions administratives de Geonames qu'il faudrait compléter par les régions historiques. Pour la France, elles sont en général très simples (Bretagne, Aquitaine, Pays Basque, Picardie, Lorraine, Alsace, etc.). Les départements peuvent aussi être utilisés pour les mettre en correspondance avec ces régions historiques. Par exemple, la commune de Mons est citée dans six départements (Charente, Charente-Maritime, Gard, Hérault, Puy de Dôme et Var). Il est donc utile d'avoir le nom du département dans la phrase afin de retourner la commune correcte.

Dans le cas où nous ne pouvons pas appliquer les relations d'inclusion, il convient de

calculer des distances entre lieux et de les comparer. Lorsqu'il y a ambiguïté sur l'un ou l'autre lieu voire sur les deux, il convient de calculer le plus rapidement possible quelle est la distance minimale entre chacune des options de façon à choisir le couple de lieux séparés par la plus petite distance. Pour minimiser les calculs, une position GPS unique sera utilisée même si une zone pourrait être défini selon un quadrilatère. Dans le cas d'une ville, cela ira beaucoup plus vite. Cela demande donc de relever une position centrale comme fournie par Geonames. Enfin, il arrive parfois qu'il n'y ait pas d'ambiguïté mais le nom donné n'est pas forcément connu. Nous allons alors le situer près d'un lieu que tout le monde connaît grâce à des prépositions de type « au Nord de », « près de », « vers ». Notons que les lieux-dits peuvent être identiques dans plusieurs points du territoire. Dans ce cas, la proximité avec un lieu connu permet de résoudre l'ambiguïté.

## 7.2 Émission d'alerte

Avec l'avènement du Web et de l'*open data*, Twitter représente une des sources externes les plus utilisées, mais nécessitant une adaptation des ressources propres à tout système traitant habituellement des textes de grande taille pour rechercher des informations particulières ; d'autant plus lorsque une fonctionnalité du système d'information est la détection d'événements à des fins de déclenchement d'alerte de danger (par exemple, un feu, un tremblement de terre, etc.).

Afin d'affiner la pertinence du déclenchement d'alerte, notre deuxième objectif, nous avons complété une ontologie existante dirigée par un corpus *ad hoc* au contexte des événements afin d'inférer des motifs (dits *patterns*) de « cause à effet ».

### 7.2.1 Regroupement de messages

Dans le cadre du projet "*Safecity*", GEOLSemantics réalise un système capable de détecter dans les réseaux sociaux (en particulier Twitter) des événements (incendies, accidents, inondations, etc.) qui pourraient nécessiter l'intervention des autorités locales (pompiers, ambulance, police, etc.), et de caractériser ces événements en particulier avec leur localisa-

tion précise. À la demande des partenaires, nous devons envoyer l'alerte via Kafka [1] dans un contexte de *Stream processing*.

## 7.3 Adaptation de la chaîne de traitement GEOLSemantics

Le système de détection d'événements à des fins d'émission d'alertes à partir des réseaux sociaux développé par GEOLSemantics consiste en une analyse des tweets émis à propos d'une zone géographique précisée par le client de l'application. La chaîne de traitement GEOL présentée dans la sous-section 1.1.1 se fonde donc sur une ontologie et des composants interconnectés qui vont devoir être adaptés pour le traitement des tweets.

Nous avons vu dans la partie 1.1.1 du chapitre 1 que GEOLSemantics dispose déjà d'un outil d'analyse linguistique profonde et d'un outil d'extraction de l'information. Cet outil a connu plusieurs évolutions, notamment pour le traitement du chinois afin de gérer la problématique de l'absence des espaces à l'écrit et la désambiguïsation des mots qui, pour une même graphie, ont des sens différents qui ne peuvent être révélés que par l'intonation de l'orateur. Pour cela, l'outil GAL a vu les modules de tokenisation, catégorisation, lemmatisation et désambiguïsation se transformer en

— une analyse typographique ;
— une analyse morphosyntaxique ;
— un traitement des mots inconnus ;
— une désambiguïsation ;
— une analyse syntaxique profonde ;
— une identification des entités nommées ;
— une résolution des anaphores.

Les outils de GEOLSemantics étaient jusque-là conçus pour des textes journalistiques ou juridiques, traitant des phrases bien structurées et sans trop de fautes d'orthographe. Avec

1. https ://kafka.apache.org/

Twitter et ses spécificités d'écriture, des modifications importantes sont nécessaires. C'est ce que nous allons présenter dans les sections suivantes.

### 7.3.1 Prétraitement des tweets

En plus des dictionnaires argotiques, GEOLSemantics a développé un outil permettant la détection de mots dans un grand flux de données grâce à un dictionnaire arborescent dont chacun des nœuds contient une table de hachage indexant ses branches.

Le premier nœud contenant dans sa table le premier caractère de chaque mot est le nœud racine (dit "ROOT"). Un nœud finissant un mot est doté d'un champ booléen *EndOfWord* permettant de déterminer s'il s'agit d'une fin de mot ou non.

Le dictionnaire est d'abord fléchi pour obtenir toutes les formes de chacun des mots. Ce dictionnaire est ensuite phonétisé, puis regénéré en parsant le fichier d'entrée caractère par caractère en suivant l'arborescence déjà générée créant une nouvelle branche (une nouvelle entrée dans la hashtable des branches d'un nœud) si un caractère ne correspondant pas à une branche est détecté (cf. figure 7.2).

Le fait de changer les mots de l'argot vers le français avec le dictionnaire argotique et de corriger les fins de mots avec l'outil de phonétisation permet de constituer une entrée qui garantit une analyse sémantique plus propre.

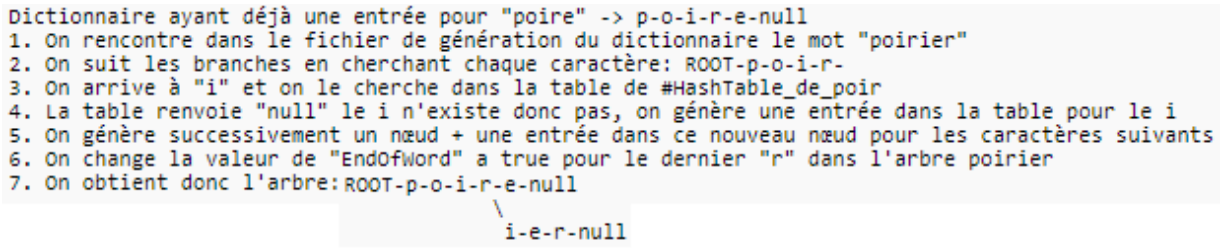



FIGURE 7.2 – Processus d'écriture de *Poire* et ses déclinaisons dans le dictionnaire

En plus du prétraitement orthographique et phonétique des tweets, GEOLSemantics a fait le choix, pour le moment, de ne pas utiliser les émoticônes et les retweets dans le système actuel. Bien que ces deux éléments peuvent aider dans le calcul du degré de véracité du tweet contenant un événement, l'utilisation d'une émoticône, par exemple, peut nous

donner des informations sur les sentiments du twittos. Nous avons choisi de les intégrer dans un second temps, afin de ne pas perturber la syntaxe du tweet car leur prise en compte est primordiale.




(a) Un état festif




(b) Un état de choc

FIGURE 7.3 – Tweet avec émoticônes

Les exemples de la figure 7.3 contiennent tous les deux le même tweet mais à la lecture des émoticônes, on comprend que le tweet 7.3a parle d'une fête qui se passe dans la maison d'à côté, puisque « *C'est le feu* » est une expression idiomatique pour exprimer une ambiance de fête. Par contre le tweet 7.3b traduit l'état de choc dans lequel se trouve le twittos et peut-être probablement un tweet sur un incendie.

Il existe deux types d'émoticônes à savoir imprimables et non imprimables[2] (ou non-bmp). Les caractères non imprimables sont remplacés directement par un espace lors du pré-traitement du tweet. En ce qui concerne les caractères imprimables, ils sont ajoutés dans un dictionnaire de pré-traitement, appelé *dicopretraitement*, et remplacés par un espace. Le dicopretraitement a comme clé l'émoticône originale et en valeur sa chaîne de remplacement selon l'algorithme 15. Le dicopretraitement sert aussi dans le pré-traitement des mots puisqu'il contient aussi une liste de mots du langage SMS et leurs chaînes de remplacement (cf. algorithme 15). Par exemple, le texte « jsui pas » correspond à « je ne suis pas ».

Les caractères imprimables sont ajoutés au fur et à mesure des tests à chaque apparitions

2. what-are-the-most-common-non-bmp-unicode-characters-in-actual-use

dans les tweets. Pour le premier corpus contenant des émoticônes, nous avons utilisé des dictionnaires d'émoticônes disponibles sur internet comme getemoji [3] ou emojipedia [4].

**Algorithme 15 :** Remplacement des émoticônes dans un tweet

**Data :** *tweet* : tweet à traiter ; *dict* : dictionnaire dicopretraitement
**Result :** tweet sans émoticône
**if** émoticônes non imprimables **then**
*non_bmp_map* ← dict.fromkeys(range(0x10000, sys.maxunicode + 1), " ")
*tweet* ← tweet.translate(*non_bmp_map*)
**else**
*tweet* ← tweet.replace(dict[original], dict[remplacement])
**end if**
renvoyer *tweet*

Un retweet est une nouvelle publication d'un tweet en citant le twittos original. La fonction *Retweet* de Twitter permet de partager un Tweet pour qu'il soit visible sur la page Twitter de celui qui a retweeté. On peut retweeter nos propres Tweets ou les Tweets de quelqu'un d'autre. Parfois, les gens saisissent "RT" au début d'un Tweet pour indiquer qu'ils publient à nouveau le contenu de quelqu'un d'autre. [5]

Le retweet est aussi un autre indicateur de sentiments et de véracité d'un événement qui n'a pas été pris en considération par GEOLSemantics En effet si un tweet parlant d'un événement et retweeté en masse c'est une indication que l'événement est vraiment réel et qu'il touche beaucoup de personnes (un retweet est unique, chaque retweet représente une personne).

Mais il faut retenir que ceci n'est pas toujours vrai, puisque comme ça que se répandent les rumeurs : les gens ne vont pas toujours vérifier les informations qu'ils retweetent. calà à constitué une autre raison pour ne pas prendre cet indicateur en considération.

L'idée de GEOLSemantics est de faire un système qui détecte l'événement à partir du premier tweet, bien avant qu'il soit un événement connu par la twittosphere. En utilisant les retweets comme indicateur cet objectif serait de facto abandonné.

Les retweets ne sont d'ailleurs même pas récupérés lors de la requête de récupération, en

3. `https://getemoji.com`
4. `https://emojipedia.org`
5. `https://help.twitter.com/en/using-twitter/retweet-faqs`

utilisant l'option (-rt).

Mais nous gardons en perspective de regarder ce que peut apporter l'étude des retweets plus tard, surtout que plusieurs études récentes reposent sur cette fonctionnalité.

Chen *et al.* [2020] ont étudié les sentiments et les schémas des retweets dans les zones sinistrées et les zones non sinistrées à différents stades de l'ouragan Harvey [6]. L'étude a confirmé l'effet des sentiments (positifs, négatifs), de la zone (la zone frappée par l'ouragan/les zones autour) et de l'heure (avant, pendant, après l'ouragan) sur la vitesse de retweet a été confirmé.

Wang *et al.* [2020]présentent une analyse des messages d'alerte aux risques envoyés via Twitter lors de l'ouragan Irma, qui a frappé Key West, en Floride et en Floride continentale le 10 septembre 2017. L'étude a analysé 497 tweets envoyés par six organisations de gestion des urgences. Cette étude a confirmé que les tweets contenant des éléments d'avertissement (danger, orientation, lieu, heure et source) ont un effet positif sur le nombre de retweet.

Bhowmick [2020] ont présenté un modèle temporel des retweets des appels à l'aide pour maximiser la diffusion d'informations sur Twitter.

### 7.3.2 Adaptation de GAL au traitement des tweets

Dans le chapitre 1, nous avons vu que notre système se fonde sur une analyse linguistique profonde (cf. module GAL de la figure 5.3), réalisée grâce à un moteur développé en interne chez GEOLSemantics [7], afin de répondre aux exigences spécifiques à nos besoins dans des contextes variés et pour des langues aux codifications très différentes : alphabet non latin, système d'agglutination, création de mots, structures syntaxiques spécifiques, traitement de textes pauvres typographiquement comme les SMS, etc. Nous analysons déjà le français, l'anglais et l'arabe. Bien évidemment, chaque nouvelle langue à traiter demande une adaptation des dictionnaires et des règles linguistiques et d'extraction sémantique.

Les tweets, quant à eux, sont rarement écrits dans un langage soigné mais remplis d'abré-

6. `https://www.lexpress.fr/actualite/monde/amerique-nord/l-ouragan-harvey-inonde-le-texas-fin-aout-1938231.html`
7. http ://www.GEOLSemantics.com/index.php/fr/

viations, sigles, argot, fautes d'orthographe, etc., sauf lorsqu'ils sont émis par des autorités, des personnes influentes issues d'un certain milieu, ou encore à des fins publicitaires (politiciens *vs.* footballeurs). Dans les autres cas, ils sont généralement pauvres en diacritiques, et en majuscules. On y recense beaucoup d'abréviations, d'émoticônes, d'url, de sigles et d'argot. Ils contiennent souvent des fautes d'orthographe, ainsi que des structures syntaxiques simplifiées. Il a donc fallu intégrer des heuristiques particulières dans notre moteur d'analyse linguistique.

**Le processus GAL.** Les éléments du processus GAL adaptés pour les tweets sont la tokenisation, la lemmatisation et la désambiguïsation, composants vus en détail dans le chapitre 1.

**La tokenisation** est l'occasion de reconnaître, en plus des mots, certaines unités remarquables par leur forme : numéros de téléphone, courriels, adresses internet, etc. Nous pouvons aussi ajouter à la demande la reconnaissance de nouvelles entités, propres à certains domaines, comme le numéro de sécurité sociale par exemple.

Pour le traitement des tweets, nous avons ajouté la reconnaissance des mots-dièse (dits *hashtags*) et des pseudonymes précédés d'une arobase.

Ainsi sont-ils reconnus comme un seul token. Le dièse et l'arobase sont supprimés au profit d'un type permettant de conserver l'information qu'il s'agissait d'hashtags ou de "mots-arobase".

Cela simplifie l'analyse lorsque certains hashtags sont utilisés comme des mots appartenant à la phrase (par exemple, *"Une habitante se plaint de la présence des dépôts sauvages et des rats à #Courbevoie"*).

**La lemmatisation** : bien que beaucoup utilisent la notion de *Stemming* dans un contexte de tweets [Shoukry and Rafea, 2012], nous avons opté pour la lemmatisation, puisque GEOLSemantics possède des dictionnaires de formes fléchies, ainsi qu'une analyse typographique des mots inconnus. Compte-tenu du fait que les fautes d'accent et de casse sont assez courantes, les dictionnaires contiennent aussi les formes *désaccentuées* des formes fléchies, c'est-à-dire les formes sans majuscule ni accent. Ainsi, lorsqu'un mot n'est pas trouvé dans le dictionnaire, il est désaccentué et recherché

dans les formes désaccentuées. Par exemple, si on considère la phrase *"Evidemment, je le savais"*, le mot *"Evidemment"* n'existe pas dans le dictionnaire. Nous le désaccentuons afin d'obtenir *"evidemment"*. Nous vérifions ensuite si cette forme désaccentuée existe dans le dictionnaire ; si c'est le cas : *"evidemment"* est lié à la forme accentuée *"évidemment"* qui est un adverbe. Cette étape prévoit aussi un découpage des mots à tiret tels que « vas-tu », et des mots collés, système de construction de mots peu utilisé en français mais beaucoup utilisé dans d'autres langues, allemand et arabe notamment.

Le traitement des tweets a demandé des adaptations à plusieurs niveaux à savoir :

— la mise en place d'un *dictionnaire spécialisé* contenant les mots argotiques, les abréviations et les sigles les plus courants, par l'analyse semi-automatique d'un corpus de tweets et par une annotation manuelle de tous les mots non reconnus par notre système mais ayant une forte occurrence dans l'ensemble du corpus. Cela nous a permis d'ajouter 2179 entrées à notre dictionnaire de français ;

— le *traitement grammatical contextualisé* des fins de mots phonétiquement identiques : les fautes grammaticales les plus courantes en français se trouvent à la fin des mots : oubli du *s* du pluriel ou de la deuxième personne du singulier des verbes, confusion entre l'infinitif et le participe passé, etc. Cela nous permet de corriger selon le contexte certaines fautes grammaticales très courantes, alors que, habituellement, des mesures comme la similarité syntaxique Kerdjoudj [2015], traitent ce genre d'écart hors contexte [Trouvilliez, 2013].

  Nous avons donc modifié la construction des dictionnaires afin que les formes souvent confondues renvoient à la fois aux interprétations exactes du mot (s'il a été correctement écrit) mais aussi aux interprétations correspondant à une faute grammaticale. Ainsi, « mangé » renverra à la fois au participe passé du verbe « manger », et à son infinitif « manger ». Cela nous permet de corriger la déclinaison du mot selon le contexte trouvé, chose irréalisable juste en utilisant des mesures de similarités syntaxiques ;

— le remplacement de la désaccentuation par une *phonétisation* : outre l'oubli ou la mauvaise utilisation des accents, certains mots sont mal orthographiés. Nous

avons donc remplacé la désaccentuation par une phonétisation afin de rapprocher les mots lorsqu'ils contiennent des fautes d'orthographe. Par exemple, pour l'analyse de « Gros bazarre sur la #ligneR », après phonétisation le mot « bazarre » aura deux interprétations : *bazar* nom singulier, ou *bazars* nom pluriel. Nous allons d'abord chercher "bazarre" dans le dictionnaire sans le trouver. Nous allons ensuite chercher avec sa phonétisation "bazar", et nous allons trouver deux interprétations possibles soit "bazar" nom singulier, soit "bazars" nom pluriel ;

— le *découpage des mots collés de type hashtags*, en suivant la casse : comme signalé ultérieurement, un de nos algorithmes permet de découper les mots inconnus avec ou sans tiret, soit avec des affixes, soit avec plusieurs mots du dictionnaire. Pour les mots de type hashtags et arobase, s'ils contiennent des majuscules, nous découpons selon ces dernières afin de limiter les possibilités par la suite. Par exemple, dans un tweet tel que « La #TableRondeRATP est ouverte », le mot hashtag *#TableRondeRATP* permettra d'obtenir un découpage en *Table Ronde RATP*. De plus, les tweets contiennent souvent des mots collés, soit à cause de fautes de frappe, soit parce qu'ils traduisent un état émotionnel particulier (joie, colère, etc.). Le découpage des mots inconnus en mots collés est donc plus utilisé dans ce contexte.

**La désambiguïsation** est effectuée grâce à une méthode statistique fondée sur des trigrammes et des bigrammes de catégories établis à partir de corpus étiquetés. Nous avons développé un corpus d'apprentissage spécifique aux tweets afin de prendre en compte les structures syntaxiques particulières qu'ils contiennent.

Après la désambiguïsation, nous établissons les **relations syntaxiques** entre les différents éléments de la phrase (nom-adjectif, nom-complément de nom, mais aussi agent-action, action-objet, etc.), et détectons les **entités nommées (EN)** en nous fondant sur des règles linguistiques permettant de traiter les temporalités (déterminer les temps et modes verbaux), les modalités (doute, volonté, souhait, etc.), les modes actifs et passifs (tous deux représentés de la même manière) et les négations.

Pour les tweets, nous avons été amenés à simplifier cette partie, et à y ajouter

des règles pour traiter les structures syntaxiques simplifiées qu'ils contiennent. Par exemple, le tweet « Quasiment 30 minutes pour faire Châtelet Denfert. » ne contient pas de verbe principal. Nous établissons l'ensemble des relations suivantes :

— Mesure : minute 30 ;

— Adjectif-Adverbe : 30 quasiment ;

— NomPropre-NomPropre : Châtelet Denfert ;

— Verbe-Objet : faire Châtelet ;

— Verbe-ComplémentCirconstantielDeTemps : faire minute.

L'analyse linguistique nous a permis de lemmatiser et catégoriser les mots des tweets. Elle nous a aussi permis d'établir des relations entre les mots en les généralisant, afin de simplifier l'étape suivante, c'est-à-dire l'extraction des événements et des informations qui leur sont propres. Aussi, notre système géolocalise les événements par le renforcement de la reconnaissance des entités nommées de lieux en faisant appel à un système d'information géographique (cf. GEOLsig dans la figure 5.3), créé à partir de fichiers de l'*open data* local et d'une extraction de l'*open data* global *Openstreetmap*[8].

### 7.3.3 POSTgal

L'analyse linguistique de GEOLSemantics (GAL) permet de collecter des noms de lieux. Cet outil repose sur un dictionnaire de noms de lieux connus. Si un nom de lieu est dans une phrase et n'est pas dans le dictionnaire, il est étiqueté en tant qu'EN inconnue (notée *inc*). Une étape de désambiguïsation s'ensuit afin d'attribuer une autre étiquette à l'Entité Nommée inconnue (*i.e. loc, pers, org* pour respectivement lieu, personne ou organisation), en fonction de sa position dans la phrase et de ses relations avec les mots l'avoisinant. Par exemple, dans la phrase *Je vais à Kendira*, Kendira est absent du dictionnaire, mais étiqueté *EN loc* grâce au verbe aller. Bien que l'étape de désambiguïsation linguistique permette de collecter plus de lieux, elle reste très stricte en raison du nombre d'erreurs potentielles. Au final, il est préférable d'avoir plus d'*EN inc* que d'EN mal identifiées.

8. https ://www.openstreetmap.org/

L'utilisation d'un outil de vérification de lieux améliore le taux de désambiguïsation. Avec ce nouveau système, nous interrogeons une base de données de lieux avec les *EN inc*. Si ce nom inconnu est présent dans la base, on peut le réinjecter dans le GAL sous une forme reconnaissable en tant qu'*EN loc*. Pour cela, nous avons établi une liste de règles linguistiques qui augmente la permissivité, et permet à plus de noms de lieux probables d'être collectés. Ces règles vont également permettre de compléter les noms de lieux et de potentiellement les regrouper avec des éléments les entourant. Par exemple, *Je suis au MK 2 de Bercy* devient *Je suis au MK 2 de Bercy*; après complétion, *MK* est réuni avec *2*, puis *MK 2* avec *de Bercy*). En outre, beaucoup de lieux intègrent des noms de personnes. Là encore nous pouvons utiliser des règles linguistiques pour vérifier si une *EN pers* détectée n'est pas, en fait, un lieu. Il faut s'assurer que tous les composants du nom propre sont présents dans le lieu retenu afin d'éviter des confusions. Par exemple, si l'on recherche *Nicolas Sarkozy*, il faut que *Nicolas* et *Sarkozy* soient présents dans le résultat au risque de le rapprocher de lieux se nommant tout simplement *Nicolas*.

La figure 7.4 présente l'architecture de ce système. Il s'agit de l'ensemble des traitements effectués après l'analyse linguistique, d'où le nom *POSTgal*.

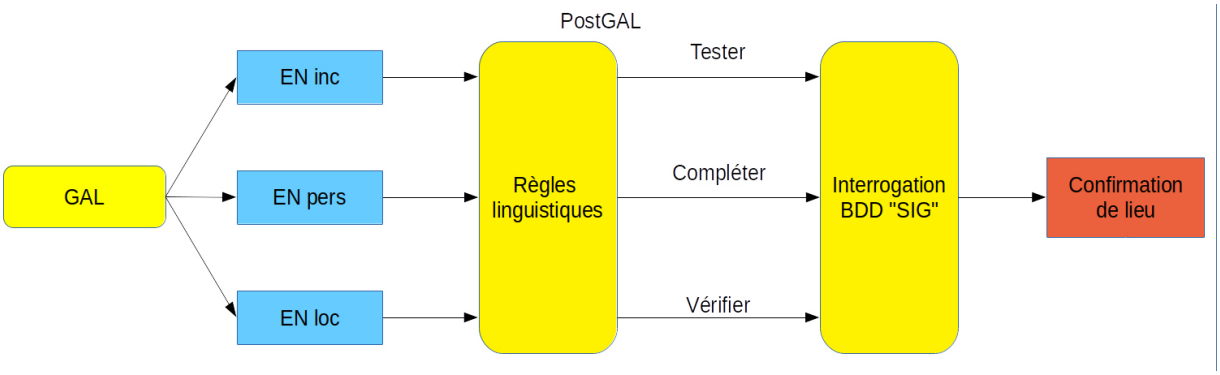



FIGURE 7.4 – Organisation du traitement POSTgal

### 7.3.4 GEOLsig

Comme mentionné en section 7.3.5, les événements découverts (grâce au module GEX) sont insérés dans une base et indexés par Elasticsearch. Ils y sont représentés au format JSON suivant une hiérarchie établie à laquelle nous rajoutons plusieurs champs de date.

L'insertion dans Elasticsearch nous permet par la suite d'exécuter les requêtes pour la fusion d'événements, qu'il s'agisse de collecter, ajouter ou supprimer des entrées. Le contenu de la base est aussi utilisé pour la visualisation finale ou soumis à d'autres modifications. Un nettoyage de la base de données est réalisé régulièrement dans le but de supprimer les événements (dits périmés) dont la date de validité est dépassée. Ainsi, la base conserve une taille réduite, assurant de meilleures performances pour les divers programmes l'utilisant. Par soucis d'anonymat, chaque événement est caractérisé par un identifiant unique attribué avant son insertion dans la base et généré par nos soins.

### 7.3.5 Event manager

Il est possible que des événements extraits soient similaires. Pour cette raison, chaque événement nouvellement extrait doit être comparé avec ceux précédemment collectés. S'ils s'avèrent similaires, ils sont alors fusionnés. La fusion est effectuée lorsqu'il y a une intersection temporelle et spatiale entre les événements, et que les actions qu'ils représentent sont identiques. Afin de vérifier une telle similarité, nous procédons en deux temps. Une requête permet de récupérer dans notre base de données les événements ayant des actions identiques et dont les intervalles de validité se chevauchent, puis l'intersection spatiale est calculée en utilisant une bibliothèque Java appropriée [Davis, 2012]. Nous aurions également pu ajouter une clause à notre requête pour vérifier ce dernier point, mais le système d'indexation Elasticsearch [9] que nous utilisons ne permet pas de calculer l'intersection entre les différents types de surfaces directement [10], chose nécessaire pour la représentation fusionnée.

Lorsqu'ils sont regroupés, les événements conservent la liste des tweets qu'ils représentent. Les dates et les lieux doivent alors être précisés. Pour les dates de début et de fin d'événement, nous considérons les valeurs les plus anciennes comme étant les plus précises. En ce qui concerne les lieux, l'intersection des représentations est adoptée, et dans le cas d'événements localisés par des points, nous effectuons une vérification de distance à la place de l'intersection, en utilisant une valeur de référence paramétrable. Nous considérons qu'un

9. voir www.elastic.com
10. https://stackoverflow.com/questions/59650375/elasticsearch-geo-shape-query

point est toujours jugé plus précis par rapport à un polygone ou un multipoint. Pour la comparaison d'événements représentés chacun par un point, si la distance les séparant est inférieure à la valeur de référence, nous considérons qu'il y a intersection, et utilisons le centre du segment formé par les deux points comme localisation dans la fusion des événements.

Lors de la fusion, le reste des métadonnées de l'événement généré provient de l'événement le plus ancien. Néanmoins, puisque chaque événement contient la liste des tweets qui l'a généré, l'ensemble des informations originelles reste disponible dans le résultat final pour tout utilisateur. Nous avons appelé le composant responsable de ce traitement l'*Event Manager* (cf. figure 5.3).

#### Enrichissement temporel d'événement

Lorsque l'émetteur du message ne précise aucune date et que les verbes indiquent un événement en cours, nous attribuons à l'événement la date d'émission du tweet. En effet, la date du tweet qui est toujours présente dans les métadonnées est une garantie que l'événement sera toujours identifié dans le temps. Si, par la suite, dans son tweet, l'utilisateur apporte une information qui peut changer la valeur du temps, cette dernière devient prioritaire. Cette information peut être :

— relative : par exemple, dans le tweet *hier, j'ai entendu des coups de feu*, la date de l'événement sera alors la date du tweet - 24 h ;

— absolue : dans l'exemple *le vendredi 24/12/2020, j'organise une rave partie*, l'événement sera associé à la date citée dans le tweet et non à la date d'émission du tweet (qui sera donc ignorée).

Cette datation est très importante car, lors du regroupement de tweets décrivant un même événement, nous arrivons à déterminer une chronologie des tweets et donc à déterminer le début de l'événement. Par exemple, après regroupement des tweets *j'ai entendu des coups de feu, il y a une 1/2h* (noté $T1$) et *j'entends des coups de feu* (noté $T2$) sur une même zone géographique, nous savons que la date d'émission de $T2$ est plus proche du début de l'événement que celle de $T1$.

**Enrichissement lié à la localisation de l'événement**

Après l'extraction d'information fondée sur les relations syntaxiques, le manque de structures grammaticales correctes dans les tweets nous a incité à enrichir l'événement par des informations extraites mais non liées syntaxiquement. Par exemple, dans le tweet *« Alarme incendie... CNIT. »*, nous avons bien extrait un événement *incendie* et un lieu *CNIT* mais les deux éléments ne sont pas liés par une relation syntaxique lors de l'analyse linguistique. De ce fait et compte tenu des caractéristiques du tweet, nous avons décidé d'enrichir l'événement de la manière suivante : s'il y a un seul événement et un seul lieu dans un tweet, le lieu est la localisation de l'événement.

Bien qu'« acte de violence » ne fasse pas partie des événements à traiter dans le projet *Safecity*, pour les besoins de la démonstration, nous avons choisi de le traiter par les exemples visibles dans les figures 7.5 et 7.6, étant donné que c'est un exemple clair exprimant le déplacement d'un événement dans l'espace.

Dans la figure 7.5, l'acte de violence se passe à côté du marché de Noël au temps $T$. Mais 2 minutes plus tard, un tweet apporte plus de précision à cet acte de violence puisqu'il précise que c'est devant l'entrée 2 du marché de Noël.

Les deux tweets proches dans l'espace et dans le temps sont regroupés (voir annexe, code Json pour le regroupement de tweets 7.5).

Nous avons suivi le même raisonnement pour enrichir la localisation d'un événement que celui lié à sa temporalité.

Le tableau 5.1 représente la distribution des événements selon leur durée. Cette dernière va permettre de regrouper certains tweets.

Une première approche pour trouver la durée de chaque événement consistait à récupérer des tweets sur des événements célèbres et calculer la moyenne des durées entre le premier tweet qui en parle et le dernier tweet qui apporte une information supplémentaire.

La tâche était compliquée surtout qu'elle s'ajoutait à celle de récupérer tous les tweets. Nous avons donc décidé de supposer des durées à titre expérimental et de corriger le laps de temps prévu au cours des tests. Au bout d'un certain nombre d'expérimentations, nous

**Un homme avec une veste grise et une matraque essait de rentrer de force dans le marché de Noël place Masséna.**

**Type de l'évènement :** Acte de violence (armé)
**Lieu de l'évènement :** marché de noël
**Date de l'évènement :** 22-09-2020 à 20:40:01

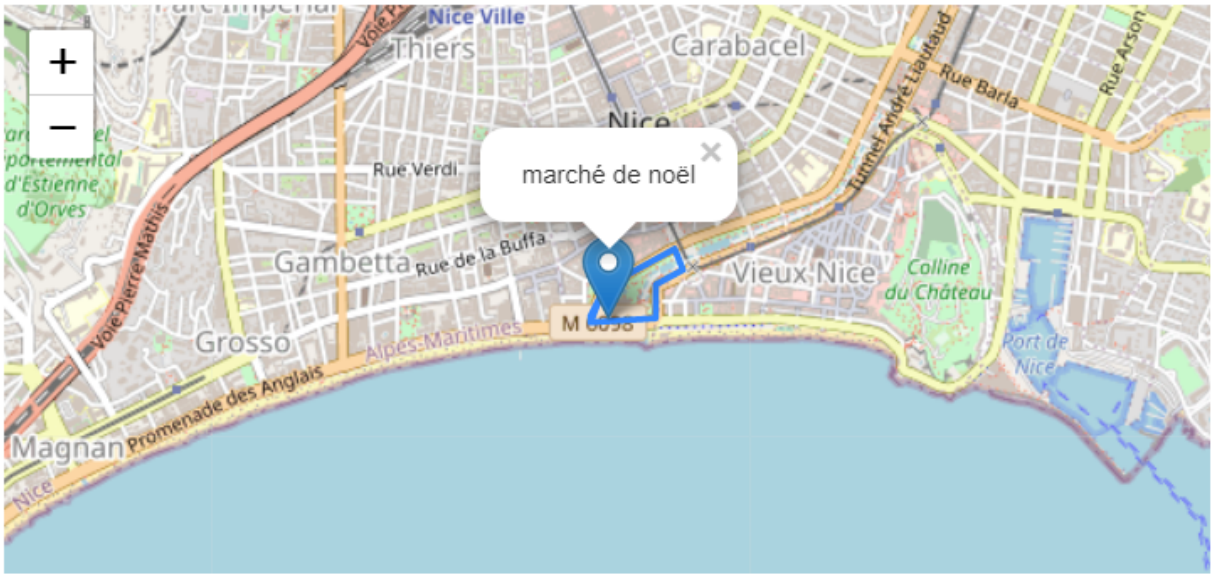



FIGURE 7.5 – Premier tweet lié à l'événement « acte de violence » autour du marché de Noël

sommes arrivés aux durées proposées dans le tableau 5.1.

Par exemple, si on a un tweet qui parle d'une bagarre spécifiant un lieu donné, tous les tweets qui parlent d'une bagarre autour de ce lieu pendant 5h seront regroupés.

Le premier tweet après ce laps de temps sera considéré comme un tweet annonçant un nouvel événement et nécessitant une nouvelle alerte.

En ce qui concerne l'organisation de fêtes les tweets seront regroupés 7 jours avant les dates annoncées dans le premier tweet. Si la date annoncée dans le tweet est passée, on considère l'événement passé et les tweets sont ignorés.

Nous avons essayé de définir une durée pour chaque événement. En effet, selon Whitehead, la durée est un ensemble « à l'unisson » d'une expérience, c'est-à-dire que pour chaque « actuel » ou occurrence appartenant à un tel ensemble, les autres actuels lui sont

**Deux personnes se battent avec les personnent qui filtrent à l'entrée 2 du marché de Noël vers place Masséna**

**Type de l'évènement :** Acte de violence (bagarre)
**Lieu de l'évènement :** entrée 2
**Date de l'évènement :** 22-09-2020 à 20:42:01




FIGURE 7.6 – Deuxième tweet lié à l'événement « acte de violence » apportant une précision sur le lieu de la figure 7.5

directement « présents » [Livet, 2008].

Partant de là, nous avons décidé de considérer que la durée d'un événement sur Twitter est situé entre le premier tweet de l'événement et le dernier tweet apportant une information nouvelle sur cet événement.

## 7.4 Expérimentations

Nous avons donc réalisé un système de détection des triplets $(date, lieu, action)$ afin de détecter les événements dans les tweets. Pour vérifier la pertinence de nos apports théoriques sur la détermination au plus tôt de ces éléments d'un triplet en vue du déclenchement d'alerte dont des améliorations pour la partie écriture des règles et la détection

des lieux en ajoutant un système d'informations géographiques, nous avons procédé à des expérimentations à partir de corpus de tweets plus ou moins conséquents.

Nous allons dans ce qui suit plus particulièrement tester et comparer avec BERT les améliorations apportées à la détection des lieux sur un flux de tweets. Rappelons que notre processus de détection suit trois étapes :

— l'étape d'extension d'extraction des lieux en utilisant des règles linguistiques (noté RL dans le tableau 7.3) : cette étape consiste à renvoyer plus de lieux que ceux sortis par l'analyse linguistique GAL mais dont certains sont à vérifier,

— l'interrogation Elasticsearch sur la base de données GEOLsig (ES dans le tableau 7.3) : elle est réalisée avec les lieux renvoyés lors de la première étape sur la base de données géographique et en ne conservant que les lieux qui existent vraiment,

— la phase de regroupement de lieu engendrant le résultat final (RF dans le tableau 7.3) : le résultat final consiste à regrouper les lieux ou en annuler certains selon la présence ou non de préposition ou l'intersection entre des lieux.

Nous avons vu dans la section précédente comment collecter les tweets soit par *bounding box* soit par requête affinée. Rappelons que la récupération par *bounding box* nous a permis de récupérer beaucoup de tweets sans lieux ni événements ce qui est idéal pour ajouter du bruit et simuler un vrai flux de tweets. Rappelons aussi que les lieux qui composent la requête affinée sont des lieux qui existent à Nice issus de la liste de tous les lieux de Nice sur Openstreetmap.

Nous avons procédé à l'annotation manuelle de 20% des tweets pour vérifier la véracité des lieux indiqués. Cette annotation nous a permis de découvrir que les tweets sans lieu sont des tweets composés de lieux avec un nom ambigu comme « Le Président » qui est un arrêt de bus à Nice comme le montre la figure 7.7. En conséquence, nous avons procédé au remplacement des lieux très ambigus par la valeur *null*.

### 7.4.1 Déroulement de l'expérimentation

Le traitement du corpus lors de l'expérimentation sera légèrement différent entre GEOLsig et BERT (*cf.*7.1).

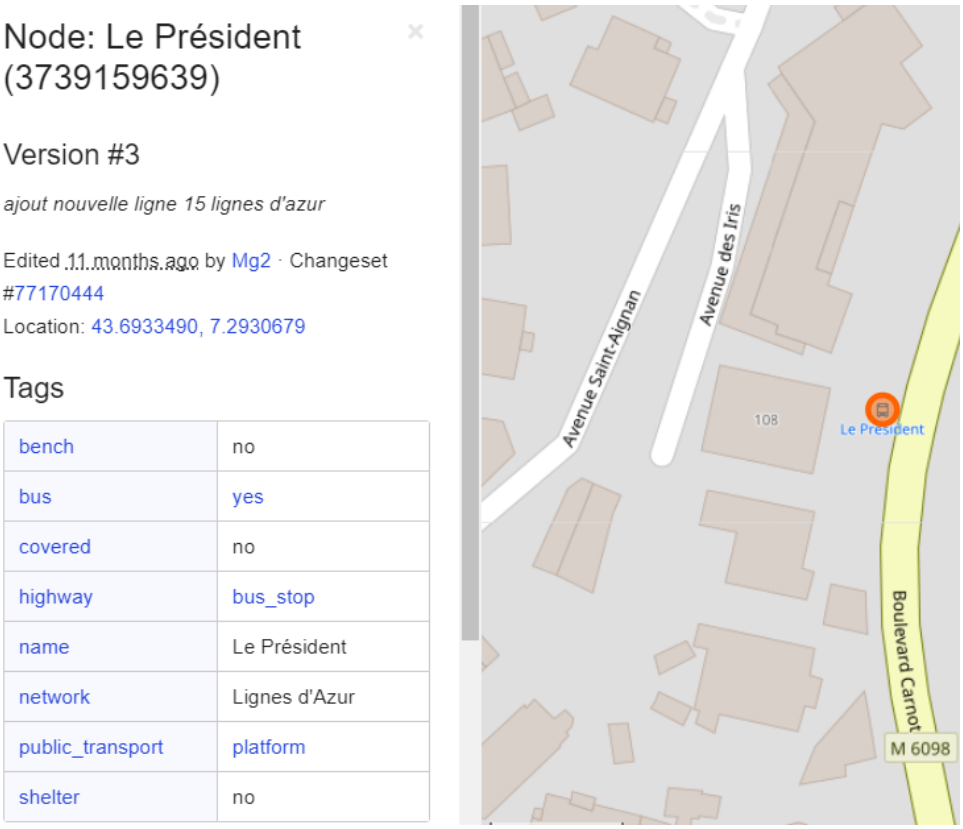



FIGURE 7.7 – Ambiguïté du terme « Le Président » pour la ville de Nice

En effet, pour GEOLsig, nous avons simulé un flux de tweets puisque notre étude a pour objectif de déterminer un événement au plus tôt. Chaque tweet va passer à travers les traitements de GEOLsig, les résultats seront stockés sans le texte du tweet. Pour pouvoir comparer les résultats en terme de précision et rappel entre GEOLsig et BERT, nous devons tout de même attendre d'avoir traité tout le corpus. Aussi, le résultat du traitement à la volée de chaque tweet par GEOLsig est regroupé dans un document.

Remarquons qu'en ce qui concerne le corpus de développement pour cette expérimentation, nous n'allons pas développer de nouvelles règles puisque nous avons déjà développé des règles pour les tests effectués sur le corpus de la première expérimentation (*cf.* 6.3).

Pour BERT, nous avons récupéré un corpus d'apprentissage et un corpus de test que nous avons dû stocker temporairement. Le corpus de test est le même que celui traité par GEOLsig. Par contre, le corpus d'apprentissage a été constitué aléatoirement des tweets récupérés par *bounding box* pour les tweets qui ne comportent pas d'événement et par requête affinée pour les événements à traiter comme indiqué dans le tableau 7.2.

| | GEOLig | BERT |
|---|---|---|
| Traitement | traitement au fil de l'eau | apprentissage puis application sur un corpus de test |
| Gestion de la donnée | corpus non stocké | corpus stocké temporairement |
| Corpus de test | 25 000 tweets | 25 000 tweets |
| Corpus d'apprentissage | 118 tweets 6.3 | 75 000 tweets |

TABLEAU 7.1 – Différences entre BERT et GEOLsig lors de l'expérimentation

Bien que pour cette expérience nous nous soyons cantonnés à récupérer des lieux, nous avons décidé de les classer selon les actions possibles car ils sont exprimés différemment selon la dépendance à une action ou une autre.

D'autre part, dans un flux de tweets standard, la majorité des tweets ne comportent pas de lieux. Collecter plus de 50% de tweets sans lieux pouvait affecter grandement l'apprentissage de BERT ; aussi nous avons décidé de diminuer à 2% de tweets sans lieux.

Dans le fichier *listeTweet* qui va nous servir pour la récupération du texte des tweets, nous avons mis aléatoirement des lignes permettant de récupérer le tweet et d'identifier son annotation. Chaque ligne contient trois éléments :

— élément 1 : l'identifiant du tweet,

— élément 2 : le lieu qui a composé la requête affinée,

— élément 3 : le lieu annoté en remplaçant par *null* les lieux qui sont ambigus.

L'algorithme 16 présente le détail du traitement lors de l'expérimentation.

En ce qui concerne BERT, nous avons utilisé le même algorithme de détection d'entité nommée que pour l'expérimentation 6.3. Les seules modifications apportées sont la suppression de la classification des événements.

### 7.4.2 Analyse des résultats

Afin de calculer la précision, le rappel et la F-mesure à partir des résultats obtenus par GEOLsig par rapport aux lieux détectés par BERT, nous avons regroupé en un seul fichier les éléments suivants : $id_tweet$, $lieu_requete$, $lieu_desambiguise$, $RL$, $ES$, $RF$, $BERT$.

| Type de collecte de tweets | Type d'action | nombre de tweets | Tweets d'expérimentation | | |
|---|---|---|---|---|---|
| | | | Corpus d'apprentissage | Corpus de test | Total |
| Par requête affinée | Incendie | 23 035 | 16 994 | 3 009 | 20 003 |
| | Inondation | 8 420 | 6 196 | 934 | 71 30 |
| | Organisation de fêtes | 28 576 | 16 261 | 3 709 | 19 970 |
| | Dépôt d'ordures | 855 | 558 | 145 | 703 |
| | Accident | 64 969 | 17 839 | 11 073 | 28 912 |
| | Bagarre | 9 662 | 7 470 | 1 220 | 8 690 |
| | Chute d'arbre | 21 580 | 5 406 | 3 626 | 9 032 |
| | Colis suspect | 3 774 | 2 898 | 262 | 3 160 |
| | Glissement de terrain | 877 | 378 | 207 | 585 |
| Par bounding box | null | 211 636 | 1 000 | 815 | 1 815 |
| | Total | 373 384 | 75 000 | 2 000 | 100 000 |

TABLEAU 7.2 – Création du corpus d'expérimentation

**Algorithme 16 :** Algorithme de procédure de l'expérimentation

**Data :** *listeTweet*, liste regroupant *id_tweet*, *lieu_requete*, *lieu_desambigu*, *corpusTest*, fichier de stockage des tweets pour tester BERT, *corpusApprentissage*, fichier de stockage des tweets pour entraîner BERT *nbTweet*, nombre de tweets récupérés
**Result :** *resultatGEOLsig* : contient *listeTweet*, *RL*, *ES*, *RF*

```
1 begin
2     for chaque tweet récupéré par id_tweet dans listeTweet do
3         if nbTweet > 25 000 // 25 000 premiers pour le corpus de test then
4             stocker le tweet dans corpusTest,
5             lancer GEOLsig et stocker RL, ES, RF dans listeTweet
6         if nbTweet > 100 000 // 75 000 suivants pour l'apprentissage then
7             stocker le tweet dans corpusApprentissage
8         nbTweet = nbTweet +1
9     ; return resultatGEOLsig
```

Notons aussi que pour cette expérimentation, nous avons élargi notre système d'information géographique à toutes les villes de France. Il est donc composé de tous les lieux de Nice en plus des noms de villes en France métropolitaine.

Le résultat de notre expérience consiste à comparer les sorties de chaque étape RL, ES, RF et BERT avec les lieux annotés *lieu_requete*, *lieu_desambiguise*. Pour ce faire, la comparaison se fait d'abord entre les sorties de l'expérience et *lieu_requete*. Si les lieux ne correspondent pas, la comparaison se fait avec *lieu_desambiguise*. Chaque comparaison devrait renvoyer un des quatre éléments VP,VN,FP,FN où :

— VP (vrai positif) : si la sortie de l'expérience correspond à un lieu annoté ;

— VN (vrai négatif) : si la sortie de l'expérience est *null* et le lieu annoté est aussi *null* ;

— FP (faux positif) : si la sortie de l'expérience est *null* et le lieu annoté n'est pas *null* ;

— FN (faux négatif) : si la sortie de l'expérience n'est pas *null* et le lieu annoté est *null*.

Les résultats de l'expérimentation sont indiqués dans le tableau 7.3.

| | RL | ES | RF | BERT |
|---|---|---|---|---|
| Précision | 50.28% | 95.37% | 85.46% | 86.79% |
| Rappel | 99.98% | 99.80% | 99.77% | 89.87% |
| F-mesure | 66.91% | 97.54% | 92.06% | 88.30% |

TABLEAU 7.3 – Résultat de l'expérimentation

Nous pouvons voir dans le tableau 7.3 que bien que l'étape 1 sur l'application des règles linguistiques (RL) donne un rappel de 99.98%, le résultat reste très faible en matière de précision (seulement 50,28%) ramenant la F-mesure à 66,91%. Ceci est dû au fait que les RL renvoient beaucoup d'éléments en tant que lieux potentiels ; en effet, la validation de la véracité de ces lieux va se faire dans l'étape suivante.

L'étape suivante, soit ES, récupère les résultats de RL et valide ou non en interrogeant un système d'informations géographiques. Ce système a vraiment amélioré la précision puisque l'on atteint maintenant 95.37% amenant la F-mesure à 97.54%.

Cela dit, notre troisième étape qui est le regroupement des lieux a fait baisser les bons

résultats obtenus précédemment ramenant la F-mesure à 92.06%. Ceci est dû à plusieurs éléments qui vont être explicités dans le tableau 7.4, Par exemple, soient les deux lieux *hameau de Serres* et *Carpentras* extraits par RL. Les deux sont validés par ES. Mais RF renvoie le résultat le plus précis puisque *hameau de Serres* est contenu dans *Carpentras*. Or l'annotation est sur le lieu le plus général, soit (Carpentras) donc *hameau de Serres* n'a pas été considéré comme un FP par le programme qui calcule la précision alors que GEOLsig a bien trouvé le bon lieu à indiquer pour le déclenchement d'une alerte. Ce genre d'erreur vient d'une mauvaise annotation expliquant des statistiques moins bonnes.

En ce qui concerne BERT, bien qu'il soit aussi bon que nos règles linguistiques (RL) pour déterminer les vrais positifs (VP) et meilleur pour les vrais négatifs (VN), il a un rappel et une précision nettement plus mauvais ; ceci est dû aux faux négatifs (FN). En effet, le fait d'apprendre sur beaucoup de tweets sans lieu, et devoir renvoyer *null* comme résultat a induit BERT en erreur. Il aurait fallu faire une classification préalablement en séparant les tweets avec actions des tweets sans actions puis entraîner BERT seulement sur les tweets avec actions. Cette méthode aurait améliorée le rappel.

De même, afin d'augmenter la précision, il aurait fallu coupler la sortie BERT avec un système d'informations géographiques. En effet, en l'absence de ce dernier, on ne peut pas vérifier si le lieu est valide ou non. La vérification de la validité du lieu aurait pu éviter des erreurs comme dans l'exemple « *Ce soir il y en a 2 qui n'iront pas ensemble à la fête des voisins : Gérard Collomb Anne Hidalgo* » où ['Collomb'] est considéré comme un lieu.

Un autre point gênant concernant BERT, c'est la difficulté d'expliquer les résultats, comme dans l'exemple précédent où ['Collomb'] a été détecté comme lieu et où on ne comprend pourquoi pas ['Hidalgo'] alors que la graphie et la position sont identiques. De même, l'exemple « *Y' a une BRANCHE QUI EST TOMBÉE SUR LA MAISON ET JE SUIS TOUTE SEULE* » où ['EST', 'TOMBÉE'] a été trouvé comme sortie mais on ne comprend pas très bien pourquoi. On peut seulement supposer qu'il a interprété *EST* comme un positionnement géographique et donc que le lieu lui-même est *TOMBÉ* (*i.e*, *à l'EST de TOMBÉ*). Ceci confirme ce que nous avions avancé lors de l'expérimentation 16 du chapitre 6 à savoir qu'il est très difficile de faire de la sémantique avec des systèmes fondés sur des statistiques.

## 7.5 Conclusion

Dans ce chapitre, nous avons abordé le suivi de messages par la contextualisation des lieux afin de déclencher des alertes pertinentes. Nous en avons déduit qu'il fallait adapter le module GAL au traitement des tweets qui ont une formalisation particulière par rapport à des documents textuels. Puis, nous avons affiné les résultats de GAL en proposant de nouvelles règles linguistiques, en créant GEOLsig qui optimise la géolocalisation des lieux trouvés après application des règles. Enfin, pour que l'alerte soit émise avec le plus de précision possible, nous avons cherché à regrouper ou fusionner des lieux indiqués pour un même événement en prenant en compte des prépositions utilisées dans les tweets.

Pour vérifier le bien fondé de notre solution, nous avons fait plusieurs expérimentations avec différents corpus qui ont été mis à disposition de la communauté scientifique. L'objectif ici était d'une part de montrer l'efficacité de la reconnaissance des lieux par deux méthodes différentes mais qui, couplées pourraient donner de meilleurs résultats. En effet, BERT a besoin d'apprendre sur beaucoup de tweets par rapport à une zone donnée. Mais le plus gênant dans notre contexte, c'est que BERT est comme une boîte noire. Nous n'avons aucune traçabilité de la façon dont les résultats ont été obtenus, et donc aucune possibilité d'explication.

D'autre part, dans le contexte de déclenchement d'alerte au plus tôt (donc a priori dans un laps de temps le plus réduit possible par rapport au premier tweet émis sur un événement), de vérifier quelle est la méthode la plus efficace d"un point de vue temporel pour détecter un événement à prendre en considération lors d'un flux de tweets produits en temps réel. Cette dernière expérimentation n'a pu malheureusement être faite par manque de temps.

| Cas de figure | Nombre de cas | Tweet | Lieu Requete | lieu désambigué | RL | ES | RF |
|---|---|---|---|---|---|---|---|
| FP-VN-VN | 10039 | Ce soir il y en a 2 qui n'iront pas ensemble à la fête des voisins : Gérard Collomb et Anne Hidalgo #FeteDesVoisins #OnATousUnVoisinQui #ParJupiter | Collomb | null | la fête des voisins, Gérard Collomb, Anne Hidalgo, A Tous Un Voisin, Jupiter | null | null |
| VP-VP-VP | 9300 | Fête des Voisins 2018 à Draguignan | Draguignan | Draguignan | Draguignan | Draguignan | Draguignan [ville] |
| VN-VN-VN | 4047 | Fête des voisins dans le centre ville | Centre Ville | null | null | null | null |
| VP-VP-FP | 1080 | Beaucoup de monde et d'ambiance à l'occasion de la fête des voisins au hameau de Serres. #Carpentras | Carpentras | Carpentras | hameau de Serres, Carpentras | Carpentras, hameau de Serres | hameau de Serres |
| FP-FP-FP | 356 | #CFLINFOSALL .Suite à un accident routier sur la N6 entre Mamer et Bertrange (direction Luxembourg), les bus de substitution de la ligne 50 seront contraints de dévier par une route alternative et ne pourront malheureusement pas desservir les arrêts . | sur la N 6 entre Mamer et Bertrange (direction Luxembourg) | N 6 | CFLINFOSALL, la N 6, Mamer Bertrange, direction Luxembourg, la ligne 50,route alternative | la ligne 50, Bertrange, la N 6 | la ligne 50, Bertrange, Bertrange |
| VP-FP-FP | 148 | Côte d'Ivoire :Grave accident à san Pedro Un minicar surchargé de fidèles chrétiens du village de Gboréké, à trois kilomètres de San Pedro, a fait un tonneau sur la voie de l'université, causant la mort deux | San Pedro | null | Côte d'Ivoire, san Pedro, village de Gboréké, San Pedro, la voie de l'université | Côte d'Ivoire, village de Gboréké | consulat de côte d'ivoire [utile], Le Village [théâtre] |
| FP-FN-FN | 12 | Aujourd'hui un proverbe Sud Africain à méditer : 'Quand un arbre tombe, on l'entend ; quand la forêt pousse, pas un bruit.' Qu'en pensez vous ? #proverbedujour #diasporasante #cameroun #sante #assurance #Yaounde #douala #Proverbeafricain | Douala | Douala | Sud Africain, proverbedujour, assurance,Yaounde, Proverbeafricain | null | null |
| FP-VP-VP | 9 | Feu de forêt à Notre-Dame-de-la-Salette : le brasier fait une victime | La Salette | null | Notre,Salette | Salette | La Salette [localité] |
| VP-FN-FN | 7 | #meteo50 VigiMeteoFrance JAUNE Vent violent et pluie-inondation Prefecture50 appelle à la plus grande prudence sur les routes Informez-vous avant de prendre la route Adaptez votre conduite aux conditions météorologiques | La Route | La Route | meteo50 Vigi Meteo France JAUNE Vent, Prefecture50, les routes Informez, route, conditions météorologiques | null | null |
| FN-FN-FN | 2 | oui oui, la rivière est énormément montée mais personne est inondé à part le gymnase, on aurait pu avoir pire | Gymnase | Gymnase | null | null | null |
| Total | 25000 | | | | | | |

TABLEAU 7.4 – Répartition des VP, VN, FP, FN selon les cas de figure

# Conclusions

*« Why are we so worried about artificial intelligence ? Surely humans are always able to pull the plug ? People asked a computer, 'Is there a God ?' And the computer said, 'There is now,' and fused the plug. »*

STEPHEN HAWKING

L'un des tableaux les plus célèbres de Vincent Van Gogh est *la nuit étoilée*. Dans ce tableau, le ciel occupe la majorité de la surface et il est composé de tourbillons rappelant les nébuleuses d'étoiles et de la lune. Il représente ce que Vincent Van Gogh a vu puis passé dans un filtre émotionnel afin de peindre ce qu'il a ressenti. Malgré les déformations, ce tableau reste une représentation assez fidèle de ce que l'artiste avait vu ; ce qui a permis aux astronomes de calculer la date et le lieu exacts de cette œuvre [Whitney, 1986].

## Synthèse des contributions

Faisant le parallèle avec le travail de ces astronomes, chez GEOLSemantics, nous avons développé une solution qui sert à trouver la date et le lieu exact des tweets, quel que soit le profil de l'utilisateur, son vocabulaire et la construction de son tweet ; bien que, contrairement à *la nuit étoilée*, ce tweet ne soit pas un chef-d'œuvre. Ce travail retrace les étapes parcourues afin de développer cette solution baptisée *GEOLsig*.

Dans le chapitre 1, nous avons présenté le contexte de l'étude, en présentant GEOLSe-

mantics et le projet *Safecity* qui est à l'origine de la demande de *GEOLsig*. Nous avons aussi repositionné notre problématique par rapport à une réflexion sur l'existant où le traitement linguistique était non adapté à la détection d'événements et plus particulièrement depuis des messages courts provenant de Twitter et de manière proche du temps réel. Pour ce faire, nous avons introduit notre définition d'un événement géo-chrono-localisé qui se compose d'une action, d'un lieu et d'une date.

Le chapitre 2 est consacré à un état de l'art sur les verrous liés à notre problématique qui sont l'analyse des réseaux sociaux ainsi que l'extraction des événements et des lieux dans ce contexte. Nous avons aussi fourni un état de l'art sur les techniques de fouille de textes et de fouilles de données, des techniques essentielles par rapport à la fouille de textes qui, dans notre cas, devient de la « fouille de messages ».

Dans le chapitre 3, nous avons fait l'état des lieux des outils déjà existants chez GEOLSemantics, résultant de trois sujets de thèses passées au sein de l'entreprise. Le système fondé sur une analyse linguistique profonde suivie d'une extraction de connaissances guidée par une ontologie devait nous servir de base pour le développement de la solution *GEOLsig*, mais en le repensant pour le traitement des données issues de Twitter qui sont différentes des données classiques (textes juridiques, journalistes, ...) traitées par GEOLSemantics.

Nous avons alors enchaîné sur la constitution d'un corpus collecté à partir de Twitter dans le chapitre 4 en présentant les méthodes de collecte des tweets et les circonstances en tenant compte de la RGPD et des règles édictées par Twitter.

Nous avons ensuite étudié notre problématique concernant la récupération des tweets sur un territoire particulier par une série d'expériences sur les corpus récupérés à travers une interface développée chez GEOLSemantics. Cela , nous a permis ainsi d'offrir à la communauté scientifique un corpus anonymisé et annoté de tweets.

Dans le chapitre 5, nous avons détaillé le système de GEOLSemantics au vu de son utilisation ou de son remplacement comme précisé dans la conclusion du chapitre 3.4. À la fin de ce travail de décorticage, nous avons décidé de ne pas remplacer le système de GEOLSemantics mais plutôt d'utiliser la sortie de ce dernier afin de développer notre solution *GEOLsig*. En effet, cette modularisation entre le système de traitement linguistique pré-

existant et notre solution permet de configurer le traitement de divers types de documents multimedia (courts ou longs), semi-automatiquement ou non, en temps quasi réel ou non, à la demande de l'utilisateur final. Cette souplesse permet une précision plus ou moins fine des résultats selon la configuration.

La sortie du système de GEOLSemantics dont nous disposons en rapport à la détection des dates et des actions étant presque satisfaisante, nous avons introduit un système d'information géographique afin d'améliorer la détection des lieux. Le système d'information géographique est proposé dans le chapitre 6. Nous avons ainsi présenté les ressources utilisées afin de le développer, et la fusion entre ce système et celui de GEOLSemantics afin de regrouper les concepts action, lieu et date nécessaires pour la création de notre notion d'événement. Afin de comparer notre solution, nous avons mis en place un système similaire mais fondé uniquement sur du *machine learning* en utilisant plus particulièrement BERT.

Enfin, dans le chapitre 7, nous sommes revenus sur quelques adaptations supplémentaires effectuées sur le système initial de GEOLSemantics afin d'améliorer la précision du triplet (action, date, lieu) d'un événement. Nous avons aussi introduit les notions d'alerte et de regroupement des messages relatifs à un même événement. Ce regroupement augmente d'une part la traçabilité pour expliquer le bien fondé de l'alerte, d'autre part de déclencher l'alerte au plus tôt.

## Futures directions

Lors de l'avancée des travaux de recherche menés au cours de ce doctorat, nous avons dû faire des choix pour répondre à la problématique posée et aux différents verrous y afférents. Cependant, un certain nombre de questions restent à résoudre qui feront l'objet de réflexions à envisager à plus ou moins long terme. En voici quelques unes qui sont loin d'être exhaustives...

**Intégration plus importante du *machine learning*** : chez GEOLSemantics, le système actuel est fortement basé sur des règles linguistiques et un raisonnement non

statistique du *machine learning* qui sont en quelque sorte des boîtes noires. Nous avons néanmoins réalisé la puissance de du *machine learning*. De ce fait, nous souhaitons l'intégrer plus en amont dans notre système afin d'automatiser l'écriture des règles d'extraction d'information. En effet, les règles d'extraction sont à ce jour écrites manuellement. Cette tâche peut-être automatisée en utilisant un outil comme BERT qui permettra de détecter par alignement des patterns de règles sur un corpus, après l'avoir entraîné sur les règles pré-existantes écrites manuellement.

**Traitement du multilinguisme** : le système doit être développé pour d'autres langues (notamment l'arabe et le chinois). Les ressources, que nous utilisons offrent aussi la translittération des lieux dans ces langues. Des essais ont été réalisés sur l'arabe au cours de la thèse [Saadane *et al.*, 2018], [Saadane *et al.*, 2017] ; malheureusement, il aurait fallu plus de temps pour les orienter vers la géo-chrono-localisation des événements et pour les adapter au langage des réseaux sociaux avec du *code-switching*.

**Prise en compte des *fake news*** : nous devons aussi travailler sur la détection de la désinformation qui est un sujet sensible qui gangrène les réseaux sociaux. Nous avons déjà établi quelques pistes au cours de cette thèse. Cependant, ce travail doit être affiné afin de rendre des résultats satisfaisant ; surtout pour un système qui est censé renvoyer des alertes correctes.

**Amélioration de la localisation** : nous avons fait une étude sur l'affichage des lieux s'ils sont combinés avec des prépositions. Ce travail doit être encore approfondi. En effet, nous avions commencé l'étude par le postulat qu'un tweet ne mentionnait au plus qu'un lieu compte-tenu que les tweets sont très courts. En réalité, un tweet peut contenir plusieurs informations sur le lieu mentionné. Il faut donc continuer l'étude des prépositions afin d'afficher le lieu le plus précisément possible.

**Augmentation de la zone d'influence de GEOLsig** : pour ce travail, nous avons utilisé un SIG autour de la zone de Nice dans le cadre du projet *Safecity* qui a servi de terrain d'expérimentation. Il s'agit donc maintenant d'élargir cette zone sur toute la France, voire d'autres pays en étudiant l'impact que peut avoir sa scalabilité et la pertinence de la géo-chrono-localisation des événements.

[( ][Intégration d'autres réseaux sociaux] : nous avons travaillé sur Twitter compte-

tenu que c'est la seule plate-forme de réseaux sociaux qui propose une API afin de récupérer les tweets et que la majorité des tweets partagés par les utilisateurs sont publics. Cependant, nous devons aussi envisager des solutions afin de pouvoir traiter en temps réel le contenu public des autres réseaux sociaux.

**Traitement de tweets multimedia** : Twitter va bientôt offrir la possibilité de tweeter en enregistrant un message. Ceci va constituer un nouveau défi car les règles de traitement seront en partie différente. Par exemple, il faudra traiter les tweets vocaux après les avoir transformés en texte afin d'avoir un système complet. Il faut aussi penser à ajouter le traitement des images, des vidéos et des émoticônes. Ces derniers sont, comme on a pu le constater, des indicateurs d'émotions pouvant nous guider dans la détection de métaphores influençant le déclenchement d'alertes ou non.

**Optimisation du déclenchement d'alerte** : TwitterBot est aujourd'hui intégré dans le flux des tweets de Twitter. Il retweete les événements en temps réel en utilisant les éléments retweetés. Nous devons améliorer notre système afin de le rendre public.

**Développement d'un système *"citizen-centered"*** dans les articles [Seffih *et al.*, 2019a] et [Seffih *et al.*, 2019b], nous avons essayé de proposer un système participatif de géolocalisation en temps réel et semi-réel fondé sur les échanges entre utilisateurs de réseaux sociaux, système capable de fonctionner dans des régions qui ne disposent pas de base de données libre de droits mais où la population est très active sur les réseaux sociaux. Ce système va permettre de transformer du texte en coordonnées GPS en utilisant les échanges de messages. L'application d'un tel système peut aller de l'implémentation d'une solution de détection de transport en temps réel à la détection des événements nécessitant une intervention.

# Annexes

## A1 - Exemple de règle pour la reconnaissance des antécédents des pronoms personnels sujet.

---

```
1
2 "granularity": "sentence",
3 "name": "rule_pronperssuj_3psf",
4 "trigger": {
5   "token": "tokens.PRONPERSSUJ_3PSF"
6 },
7 "leftContext": {},
8 "succession": [
9   {
10    "tokenC": {
11      "card": {
12        "min": "1",
13        "max": "1"
14      },
15      "token": "tokens.ANTECEDENT_SUJ_FS"
16    }
17  },
18  {
```

```
    "tokenC": {
      "card": {
        "min": "0",
        "max": "*"
      },
      "tokenNeg": "tokens.ANTECEDENT_SUJ_FS"
    }
  }
],
"actions": {
  "action": "findAntecedent",
  "params": [
    "$trigger",
    "$leftContext.succession[0].tokenC.token"
  ]
}

```

## A3 - Exemple d'un json d'un lieu à indexer.

```

"fields": {
  "_categorie": [
    "club canin"
  ],
  "_name": [
    "Royal Canin"
  ],
  "_location": {
```

```
    "type": "polygon",
    "coordinates": [
      [
        [
          43.7926831,
          7.2478894
        ],
        [
          43.7927045,
          7.248096
        ],
        [
          43.7924489,
          7.248246
        ],
        [
          43.792164,
          7.2482169
        ],
        [
          43.7920326,
          7.2478817
        ],
        [
          43.7923142,
          7.2478511
        ],
        [
          43.7926831,
          7.2478894
        ]
```

```
      ]
    ]
  },
  "_addrhousenumber": "",
  "_addrstreet": "",
  "_addrpostcode": "",
  "_city": "",
  "_country": "",
  "_address": "",
  "_is_in": [
    ""
  ]
}

```


## A3 - Code Json de regroupement de tweets.


```

"alert": {
  "identifier": "AR_0001",
  "sender": "Analyse_des_reseaux_sociaux",
  "sent": "2020-01-30T11:54:15+01:00",
  "msgType": "Alert",
  "info": {
    "category": "Safety",
    "event": "1",
    "description": "Acte de violence à entrée 2 [utile]",
    "resource": {
      "resourceDesc": "NULL",
```

```
      "mimeType": "NULL",
      "uri": "NULL"
    },
    "area": {
      "circle": "43.69515,7.268413 0.0",
      "polygon": null,
      "areaDesc": "entrée 2 [utile]"
    },
    "eventCode": {
      "eventId": "1",
      "eventDateEnd": "2020-01-30T11:57:56+01:00",
      "eventDateExpires": "2020-01-30T16:54:15+01:00",
      "tweetList": [
        {
          "tweetDesc": {
            "tweetId": "1072567093072138245",
            "tweetMessage": "Un homme avec une veste grise et une matraque
            essaie de rentrer de force dans le marché de Noël place Masséna ",
            "eventType": "Acte de violence",
            "eventSubType": "armée",
            "keywordID1": null,
            "keywordID2": null,
            "keywordID3": null,
            "hashtags": "_",
            "eventstartDate": "2020-09-2T20:40:01+01:00",
            "eventendDate": "2020-09-2T23:40:01+01:00"
          }
        },
        {
          "tweetDesc": {
            "eventstartDate": "2020-09-2T20:40:01+01:00",
```

```
            "eventSubType": "bagarre",
            "hashtags": "_",
            "tweetMessage": "Deux personnes se battent avec les personnent qui
            filtrent à l 'entrée 2 du marché de Noël vers place Masséna",
            "keywordID1": null,
            "keywordID2": null,
            "eventType": "Acte de violence",
            "keywordID3": null,
            "tweetId": "1072567093072138245",
            "eventendDate": "2020-09-2T23:40:01+01:00"
          }
        }
      ]
    }
  }
}
```

# Références


Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications : A survey. *Heliyon*, 4(11) :e00938, 2018.

Dirk Ahlers. Assessment of the accuracy of geonames gazetteer data. In *Proceedings of the 7th workshop on geographic information retrieval*, pages 74–81, 2013.

Aysha Al Nuaimi, Aysha Al Shamsi, Amna Al Shamsi, and Elarbi Badidi. Social media analytics for sentiment analysis and event detection in smart cities. *Computer Science & Information Technology*, page 57, 2018.

Nourhene Alaya Mili. *Managing the empirical hardness of the ontology reasoning using the predictive modelling*. PhD thesis, Paris 8, 2016.

Hunt Allcott, Luca Braghieri, Sarah Eichmeyer, and Matthew Gentzkow. The welfare effects of social media. *American Economic Review*, 110(3) :629–76, 2020.

Dean Allemang, Jim Hendler, and Fabien Gandon. *Semantic Web for the Working Ontologist*. ACM, June 2020.

James Allen. *Natural language understanding*. Pearson, 1995.

Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

Alexandre Andreewsky and Christian Fluhr. Expérience de constitution d'un programme d'apprentissage pour le traitement automatique du langage. In *COLING 1973 Volume 2 : Computational And Mathematical Linguistics : Proceedings of the International Conference on Computational Linguistics*, 1973.

Jeffery Ansah, Lin Liu, Wei Kang, Jixue Liu, and Jiuyong Li. Leveraging burst in twitter network communities for event detection. *World Wide Web*, pages 1–26, 2020.

Gil Appel, Lauren Grewal, Rhonda Hadi, and Andrew T Stephen. The future of social media in marketing. *Journal of the Academy of Marketing Science*, 48(1) :79–95, 2020.

Eiji Aramaki, Sachiko Maskawa, and Mizu Morita. Twitter catches the flu : detecting influenza epidemics using twitter. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1568–1576. Association for Computational Linguistics, 2011.

Imene Bensalem and Mohamed-Kiredine Kholladi. La désambiguïsation des toponymes par la densité géographique. In *International Conference on Applied Inforatics (ICAI'09)*, 2009.

Edward Benson, Aria Haghighi, and Regina Barzilay. Event discovery in social media feeds. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies-Volume 1*, pages 389–398. Association for Computational Linguistics, 2011.

Tim Berners-Lee, Wendy Hall, James A Hendler, Kieron O'Hara, Nigel Shadbolt, Daniel J Weitzner, et al. A framework for web science. *Foundations and Trends® in Web Science*, 1(1) :1–130, 2006.

Ayan Kumar Bhowmick. Temporal pattern of retweet (s) help to maximize information diffusion in twitter. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 913–914, 2020.

Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5 :135–146, 2017.

Peter K Bol. On the cyberinfrastructure for gis-enabled historiography : Space–time integration in geography and giscience. *Annals of the Association of American Geographers*, 103(5) :1087–1092, 2013.

Stefano Borgo and Claudio Masolo. *Ontological foundations of DOLCE*, pages 279–295. Springer Netherlands, 09 2010.

Willem Nico Borst and W.N. Borst. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. PhD thesis, University of Twente, Netherlands, 9 1997.

Naouel Bougriou and Soumia Milat. *Désambiguïsation des toponymes guidée par des ressources géographiques*. PhD thesis, Université Abdelhamid Mehri Constantine 2, 2016.

Bertrand Braunschweig. Artificial intelligence : Current challenges and inria's engagement. *Inria white paper*, 2016.

David A Broniatowski, Michael J Paul, and Mark Dredze. National and local influenza surveillance through twitter : an analysis of the 2012-2013 influenza epidemic. *PloS one*, 8(12), 2013.

David Brosset. *Web and wireless geographical information systems : 15th International Symposium, W2GIS 2017, Shanghai, China, May 8-9, 2017 : proceedings*. Springer, 2017.

Erick Cantú-Paz. A survey of parallel genetic algorithms. *Calculateurs paralleles, reseaux et systems repartis*, 10(2) :141–171, 1998.

Camille Cavalière, Paule-Annick Davoine, Céline Lutoff, and Isabelle Ruin. Analyser des tweets géolocalisés pour explorer les réponses sociales face aux phénomènes météorologiques extrêmes. In *SAGEO'2016*, 2016.

Soumen Chakrabarti, Martin Ester, Usama Fayyad, Johannes Gehrke, Jiawei Han, Shinichi Morishita, Gregory Piatetsky-Shapiro, and Wei Wang. Data mining curriculum : A proposal (version 1.0). *Intensive Working Group of ACM SIGKDD Curriculum Committee*, 140 :1–10, 2006.

Samprit Chatterjee and Ali S Hadi. *Regression analysis by example*. John Wiley & Sons, 2015.

Yung-Yao Chen, Yu-Hsiu Lin, Chia-Ching Kung, Ming-Han Chung, I Yen, et al. Design and implementation of cloud analytics-assisted smart power meters considering advanced artificial intelligence as edge analytics in demand-side management for smart homes. *Sensors*, 19(9) :2047, 2019.

Sijing Chen, Jin Mao, Gang Li, Chao Ma, and Yujie Cao. Uncovering sentiment and retweet patterns of disaster-related tweets from a spatiotemporal perspective–a case study of hurricane harvey. *Telematics and Informatics*, 47 :101326, 2020.

Smitashree Choudhury and Harith Alani. Personal life event detection from social media. *In Workshop on Social Personalisation*, 1210, 2014.

David Corney, Dyaa Albakour, Miguel Martinez-Alvarez, and Samir Moussa. What do a million news articles look like ? In *NewsIR@ ECIR*, pages 42–47, 2016.

H. Cramer and S. Büttner. Things that tweet, check-in and are befriended. two explorations on robotics social media. In *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 125–126, 2011.

Pascal Cuxac, Alain Collignon, Stéphanie Gregorio, and François Parmentier. From massive databases to Web of data : disambiguation and alignment of geographical entities in scientific texts. In *12ème Colloque international d'ISKO-France : Données et mégadonnées ouvertes en SHS : de nouveaux enjeux pour l'état et l'organisation des connaissances ?*, Montpellier, France, October 2019.

Richard Cyganiak, David Wood, and Markus Lanthaler. Rdf 1.1 concepts and abstract syntax, 2014 (accessed February 9, 2019).

Donald Davidson. *Essays on Actions and Events : Philosophical Essays Volume 1*, volume 1. Oxford University Press on Demand, 2001.

Martin Davis. Jts topology suite, 2012.

Jens de Bruijn, Hans de Moel, Brenden Jongman, and Jeroen Aerts. Towards a global flood detection system using social media. *EGUGA*, page 1102, 2017.

Pierre Delort. *Le Big Data :«Que sais-je ?» n° 4027*. Que sais-je, 2018.

Anneliese Depoux, Sam Martin, Emilie Karafillakis, Raman Preet, Annelies Wilder-Smith, and Heidi Larson. The pandemic of social media panic travels faster than the covid-19 outbreak, 2020.

Christine Develotte and Marie-Anne Paveau. Pratiques discursives et interactionnelles en contexte numérique. questionnements linguistiques. *Langage et société*, 160-161(2) :199–215, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*, 2018.

Martin Doerr. The cidoc conceptual reference module : an ontological approach to semantic interoperability of metadata. *AI magazine*, 24(3) :75, 2003.

Brindha Duraisamy, Jayaseelan Rathinaswamy, and Kadeswaran Sengottaiyan. Social media reigned by information or misinformation about covid-19 : A phenomenological study. *SSRN Electronic Journal*, 09 :585–602, 04 2020.

Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric P Xing. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1277–1287. Association for Computational Linguistics, 2010.

Carola Eschenbach, Christopher Habel, Lars Kulik, et al. Representing simple trajectories as oriented curves. In *FLAIRS Conference*, pages 431–436, 1999.

Atefeh Farzindar and Mathieu Roche. Les défis de l'analyse des réseaux sociaux pour le traitement automatique des langues. *Revue TAL-Traitement Automatique des Langues*, 54(3) :7–16, 2013.

Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3) :37–37, 1996.

Ronen Feldman and Ido Dagan. Knowledge discovery in textual databases (kdt). In *KDD*, volume 95, pages 112–117, 1995.

Philip Fennell. Extremes of xml. *XML LONDON 2013*, 2013.

M. Fernández-López, Asunción Gómez-Pérez, and N. Juristo. Methontology : From ontological art towards ontological engineering. In *AAAI 1997*, 1997.

Luca Foppiano and Laurent Romary. entity-fishing : a DARIAH entity recognition and disambiguation service. In *Digital Scholarship in the Humanities* , Tokyo, Japan, September 2018.

formation seo. Bert a expliqué : ce que vous devez savoir - ne pas manquer formation seo 30 jours, Apr 2020.

Alexander Frame and Gilles Brachotte. Les campagnes des compagnes, 2013.

Ivana Franić. Tesnière's «elements de syntaxe structurale» between scientific and didactic discourse «elements de syntaxe structurale» between scientific and didactic discourse. *Journal for Foreign Languages*, 3 :61, 05 2011.

Fabien Gandon and Rose Dieng. Ontologie pour un système multi-agents dédié à une mémoire d'entreprise. In *IC'2001, Ingénierie des Connaissances, plateforme AFIA'2001*, Grenoble, France, June 2001.

Aldo Gangemi and Peter Mika. Understanding the semantic web through descriptions and situations. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 689–706. Springer, 2003.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Thomas R Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2) :199–220, 1993.

Nicola Guarino. *Formal ontology in information systems : Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*, volume 46. IOS press, 1998.

Gaël Guibon, Liana Ermakova, Hosni Seffih, Anton Firsov, and Guillaume Le Noé-Bienvenu. Multilingual Fake News Detection with Satire. In *CICLing : International Conference on Computational Linguistics and Intelligent Text Processing*, La Rochelle, France, April 2019.

Nizar Habash. Four techniques for online handling of out-of-vocabulary words in arabic-english statistical machine translation. In *Proceedings of ACL-08 : HLT, Short Papers*, pages 57–60, 2008.

Mena B Habib and Maurice van Keulen. Named entity extraction and disambiguation : The reinforcement effect. *Proc. of MUD*, 2011, 2011.

Bo Han, Paul Cook, and Timothy Baldwin. A stacking-based approach to twitter user geolocation prediction. In *Proceedings of the 51st annual meeting of the association for computational linguistics : system demonstrations*, pages 7–12, 2013.

Stephen Hawking, Eddie Redmayne, Kip S. Thorne, and Lucy Hawking. *Brief answers to the big questions*. John Murray, 2020.

Marti A Hearst. Text data mining : Issues, techniques, and the relationship to information access. In *Presentation notes for UW/MS workshop on data mining*, volume 1, page 997, 1997.

Marti A Hearst. Untangling text data mining. In *Proceedings of the 37th Annual meeting of the Association for Computational Linguistics*, pages 3–10, 1999.

Marti Hearst. What is text mining. *SIMS, UC Berkeley*, 5, 2003.

Brent Hecht, Lichan Hong, Bongwon Suh, and Ed H Chi. Tweets from justin bieber's heart : the dynamics of the location field in user profiles. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 237–246, 2011.

Nathalie Hernandez. *Ontologies de domaine pour la modélisation du contexte en recherche d'information*. PhD thesis, Université Paul Sabatier-Toulouse III, 2005.

Linda L Hill. Core elements of digital gazetteers : placenames, categories, and footprints. In *International Conference on Theory and Practice of Digital Libraries*, pages 280–290. Springer, 2000.

Thi Bich Ngoc Hoang and Josiane Mothe. Location extraction from tweets. *Information Processing & Management*, 54(2) :129–144, 2018.

Frederik Hogenboom, Flavius Frasincar, and Uzay Kaymak. An overview of approaches to extract information from natural language corpora. *Information Foraging Lab*, 69, 2010.

Binxuan Huang and Kathleen M Carley. A hierarchical location prediction neural network for twitter user geolocation. *arXiv preprint arXiv :1910.12941*, 2019.

Sheng-Hao Hung, Chia-Hung Lin, and Jen-Shin Hong. Web mining for event-based commonsense knowledge using lexico-syntactic pattern matching and semantic role labeling. *Expert Systems with Applications*, 37(1) :341 – 347, 2010.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. Embeddings for word sense disambiguation : An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 897–907, 2016.

Muhammad Imran, Prasenjit Mitra, and Carlos Castillo. Twitter as a lifeline : Human-annotated twitter corpora for nlp of crisis-related messages. *arXiv preprint arXiv :1605.05894*, 2016.

Bernard Jacquemin. *Construction et interrogation de la structure informationnelle d'une base documentaire en français*. PhD thesis, PhD thesis, Université Paris III Sorbonne Nouvelle, Paris, 2003. 43, 2003.

Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering : a review. *ACM computing surveys (CSUR)*, 31(3) :264–323, 1999.

Juvénal JVCJuvénal. Interrogez efficacement vos bases de données avec elasticsearch, Apr 2020.

Edward F Kelly and Philip J Stone. *Computer recognition of English word senses*, volume 13. North-Holland, 1975.

Fadhela Kerdjoudj. *Gestion de l'incertitude dans le processus d'extraction de connaissances à partir de textes*. PhD thesis, Paris Est, 2015.

Susie Khamis, Lawrence Ang, and Raymond Welling. Self-branding,'micro-celebrity'and the rise of social media influencers. *Celebrity studies*, 8(2) :191–208, 2017.

Michel Klein. Combining and relating ontologies : an analysis of problems and solutions. In *OIS@ IJCAI*, 2001.

Jon Kleinberg. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4) :373–397, 2003.

Paul R Kroeger. Translating presuppositions. *The Bible Translator*, 70(2) :167–183, 2019.

Yury Kryvasheyeu, Haohui Chen, Esteban Moro, Pascal Van Hentenryck, and Manuel Cebrian. Performance of social network sensors during hurricane sandy. *PLoS one*, 10(2), 2015.

Olivier Rosec Kévin Vythelingum, Yannick Estève. Transcription phonétique automatique pour la synthèse de la parole. In *XXXIIe Journées d'Etudes sur la Parole (JEP 2018)*, 2018.

C Lagoze and J Hunter. The abc ontology and model, journal of digital information. *Article No. 77, 2001-11-06*, 2(2), 2001.

Thomas Laloë. *Sur quelques problèmes d'apprentissage supervisé et non supervisé*. PhD thesis, Montpellier II - Sciences et Techniques du Languedoc, 2009.

Alex Lascarides and Nicholas Asher. Segmented discourse representation theory : Dynamic semantics with discourse structure. In *Computing meaning*, pages 87–124. Springer, 2008.

Kathy Lee, Ankit Agrawal, and Alok Choudhary. Real-time disease surveillance using twitter data : demonstration on flu and cancer. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1474–1477, 2013.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT : a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4) :1234–1240, 09 2019.

Jiwei Li, Alan Ritter, Claire Cardie, and Eduard Hovy. Major life event extraction from twitter based on congratulations/condolences speech acts. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1997–2007, 2014.

Fabien Liénard and Sami Zlitni. *La communication électronique : enjeux de langues*. Lambert Lucas, 2011.

Qi Liu, Matt J Kusner, and Phil Blunsom. A survey on contextual embeddings. *arXiv preprint arXiv :2003.07278*, 2020.

Pierre Livet. La notion d'événement chez whitehead et davidson. *Noesis*, pages 217–233, 2008.

Arne Lohmann and Erin Conwell. Phonetic effects of grammatical category : How category-specific prosodic phrasing and lexical frequency impact the duration of nouns and verbs. *Journal of Phonetics*, 78 :100939, 2020.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

Cédric Lopez, Ioannis Partalas, Georgios Balikas, Nadia Derbas, Amélie Martin, Coralie Reutenauer, Frédérique Segond, and Massih-Reza Amini. Cap 2017 challenge : Twitter named entity recognition. *arXiv preprint arXiv :1707.07568*, 2017.

David C Luckham and Brian Frasca. Complex event processing in distributed systems. *Computer Systems Laboratory Technical Report CSL-TR-98-754. Stanford University, Stanford*, 28 :16, 1998.

Vida Maliene, Vytautas Grigonis, Vytautas Palevičius, and Sam Griffiths. Geographic information system : Old principles with new capabilities. *Urban Design International*, 16(1) :1–6, 2011.

Claude Martineau, Elsa Tolone, and Stavroula Voyatzi. Les entités nommées : usage et degrés de précision et de désambiguïsation. *26ème Colloque international sur le Lexique et la Grammaire (LGC'07)*, 01 2010.

Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, and Alessandro Oltramari. Wonderweb deliverable d18 : Ontology library, 2003.

Béatrice Mazoyer, Julia Cagé, Nicolas Hervé, and Céline Hudelot. A French corpus for event detection on Twitter. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6220–6227, Marseille, France, May 2020. European Language Resources Association.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation : Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305, 2017.

David D McDonald. Internal and external evidence in the identification and semantic categorization of proper names. In *Acquisition of Lexical Knowledge from Text*, 1993.

Shannon C McGregor and Logan Molyneux. Twitter's influence on news judgment : An experiment among journalists. *Journalism*, 21(5) :597–613, 2020.

Andrew J McMinn, Yashar Moshfeghi, and Joemon M Jose. Building a large-scale corpus for evaluating event detection on twitter. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 409–418, 2013.

Ida Mele and Fabio Crestani. A multi-source collection of event-labeled news documents. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 205–208, 2019.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*, 2013.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

Ahmed Mourad, Falk Scholer, Walid Magdy, and Mark Sanderson. A practical guide for the effective evaluation of twitter user geolocation. *ACM Transactions on Social Computing*, 2(3) :1–23, 2019.

Janmenjoy Nayak, Bighnaraj Naik, and H Behera. A comprehensive survey on support vector machine in data mining tasks : applications & challenges. *International Journal of Database Theory and Application*, 8(1) :169–186, 2015.

Natalya F Noy and Deborah L McGuinness. Développement d'une ontologie 101 : Guide pour la création de votre première ontologie. *Université de Stanford, Stanford, Traduit de l'anglais par Anila Angjeli. ht tp ://www. bnf. fr/pages/infopro/normes/pdf/noDevOnto. pdf*, 2000.

Peter Oram. Wordnet : An electronic lexical database. christiane fellbaum (ed.). cambridge, ma : Mit press, 1998. pp. 423. *Applied Psycholinguistics*, 22(1) :131, 2001.

Venkata Sasank Pagolu, Kamal Nayan Reddy, Ganapati Panda, and Babita Majhi. Sentiment analysis of twitter data for predicting stock market movements. In *2016 international conference on signal processing, communication, power and embedded system (SCOPES)*, pages 1345–1350. IEEE, 2016.

Sean B Palmer. The semantic web : An introduction. *VIVEK-BOMBAY-*, 14(4) :13–25, 2001.

Symeon Papadopoulos, David Corney, and Luca Maria Aiello. Snow 2014 data challenge : Assessing the performance of news topic detection methods in social media. In *SNOW-DC@ WWW*, pages 1–8. Seoul, Korea, 2014.

Terence Parsons. Events in the semantics of english. a study in subatomic semantics. number 19 in current studies in linguistics, 1994.

Tommaso Pasini and Roberto Navigli. Train-o-matic : Supervised word sense disambiguation with no (manual) effort. *Artificial Intelligence*, 279 :103215, 2020.

Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation : A survey of recent advances. *IEEE signal processing magazine*, 32(3) :53–69, 2015.

Jérémy Patrix. *Behaviour detection through semantically networked multi-agent models, applied to the situation awareness application domain, especially in asymmetric environment with uncertain and imprecise information.* Theses, Université de Caen, December 2013.

Jorge David Gonzalez Paule, Yeran Sun, and Yashar Moshfeghi. On fine-grained geolocalisation of tweets and real-time traffic incident detection. *Information Processing & Management*, 2018.

Marie-Anne Paveau. Technodiscursivités natives sur twitter. une écologie du discours numérique. *revue internationale de sciences humaines et sociales appliquées*, 2013.

Nicolas Pélissier and Gabriel Gallezot. *Twitter : Un monde en tout petit ?* Editions L'Harmattan, 2013.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove : Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Sarah Perez. Twitter officially expands its character count to 280 starting today, Nov 2017.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv :1802.05365*, 2018.

Isabelle Pierozak. Regards sur l'internet, dans ses dimensions langagières. *Penser les*, 2007.

James Pustejovsky and Jessica L Moszkowicz. The qualitative spatial dynamics of motion in language. *Spatial Cognition & Computation*, 11(1) :15–44, 2011.

James Pustejovsky. Dynamic event structure and habitat theory. In *Proceedings of the 6th International Conference on Generative Approaches to the Lexicon (GL2013)*, pages 1–10, 2013.

James Pustejovsky. Situating events in language. *Conceptualizations of Time*, 52 :27–46, 2016.

Frederik Ramm, Jochen Topf, and Steve Chilton. *OpenStreetMap : using and enhancing the free map of the world*. UIT Cambridge Cambridge, 2011.

Arnaud Rosier. *Raisonnement automatique basé ontologies appliqué à la hiérarchisation des alertes en télécardiologie*. PhD thesis, Rennes 1, 2015.

Houda Saadane, Damien Nouvel, Hosni Seffih, and Christian Fluhr. Une approche linguistique pour la détection des dialectes arabes (a linguistic approach for the detection of arabic dialects). In *Actes des 24ème Conférence sur le Traitement Automatique des Langues Naturelles. Volume 2-Articles courts*, pages 242–250, 2017.

Houda Saadane, Hosni Seffih, Christian Fluhr, Khalid Choukri, and Nasredine Semmar. Automatic identification of maghreb dialects using a dictionary-based approach. In *Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Houda Saâdane. *Le traitement automatique de l'arabe dialectalisé : aspects méthodologiques et algorithmiques*. PhD thesis, Université Grenoble Alpes, 2015.

S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3) :660–674, 1991.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users : real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860, 2010.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Tweet analysis for real-time event detection and earthquake reporting system development. *IEEE Transactions on Knowledge and Data Engineering*, 25(4) :919–931, 2012.

Mauro Scanagatta, Antonio Salmerón, and Fabio Stella. A survey on bayesian network structure learning from data. *Progress in Artificial Intelligence*, pages 1–15, 2019.

Ansgar Scherp and Vasileios Mezaris. Survey on modeling and indexing events in multimedia. *Multimedia Tools and Applications*, 70(1) :7–23, 2014.

Ansgar Scherp, Thomas Franz, Carsten Saathoff, and Steffen Staab. F–a model of events based on the foundational ontology dolce+ dns ultralight. In *Proceedings of the fifth international conference on Knowledge capture*, pages 137–144. ACM, 2009.

Michael Seadle. The great hack (documentary film). produced and directed by karim amer and jehane noujaim. netflix, 2019. 1 hour 54 minutes. *Journal of the Association for Information Science and Technology*, 2019.

Hosni Seffih, Myriam Lamolle, Aurélie Pradelles, Christian Fluhr, and Ayoub Chouchou. A new smart citizen-centered transportation application without gps. In *Proceedings of the 11th International Conference on Management of Digital EcoSystems*, pages 187–190, 2019.

Hosni Seffih, Myriam Lamolle, Aurélie Pradelles, Ayoub Chouchou, and Christian Fluhr. How ordinary city shift to smart city ? a case study "citizen-centered". In *International Conference Connected Smart Cities 2019*, pages 395–399, 07 2019.

Paul Semaan. Natural language generation : An overview. *Journal of Computer Science & Research (JCSCR)-ISSN*, pages 50–57, 2012.

Amira Shoukry and Ahmed Rafea. Preprocessing egyptian dialect tweets for sentiment mining. In *The Fourth Workshop on Computational Approaches to Arabic Script-based Languages*, page 47, 2012.

Miguel A Sicilia, Elena García, Salvador Sánchez, and Elena Rodríguez. On integrating learning object metadata inside the opencyc knowledge base. In *Advanced Learning Technologies, 2004. Proceedings. IEEE International Conference on*, pages 900–901. IEEE, 2004.

Devendra Singh Chaplot and Ruslan Salakhutdinov. Knowledge-based word sense disambiguation using topic models. *arXiv*, pages arXiv–1801, 2018.

Uthayasankar Sivarajah, Zahir Irani, Suraksha Gupta, and Kamran Mahroof. Role of big data and social media analytics for business to business sustainability : A participatory web context. *Industrial Marketing Management*, 86 :163–179, 2020.

Enrico Steiger, João Porto de Albuquerque, and Alexander Zipf. An advanced systematic literature review (slr) on spatiotemporal analyses of twitter data–technical report. *Location Based Social Networks – Systematic Literature Review*, 2014.

Zachary C. Steinert-Threlkeld. *Twitter as Data*. Elements in Quantitative and Computational Methods for the Social Sciences. Cambridge University Press, 2018.

Stefan Stieglitz, Milad Mirbabaie, Björn Ross, and Christoph Neuberger. Social media analytics–challenges in topic discovery, data collection, and data preparation. *International journal of information management*, 39 :156–168, 2018.

Rudi Studer, V Richard Benjamins, Dieter Fensel, et al. Knowledge engineering : principles and methods. *Data and knowledge engineering*, 25(1) :161–198, 1998.

Axel Suarez, Dyaa Albakour, David Corney, Miguel Martinez, and José Esquivel. A data collection for evaluating the retrieval of related tweets to news articles. In *European Conference on Information Retrieval*, pages 780–786. Springer, 2018.

C. Sun, Xipeng Qiu, Yige Xu, and X. Huang. How to fine-tune bert for text classification ? *ArXiv*, abs/1905.05583, 2019.

Shazia Tabassum, Fabiola Pereira, Sofia Fernandes, and João Gama. Social network analysis : An overview. *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery*, 8 :e1256, 04 2018.

Raphaël Troncy, Ryan Shaw, and Lynda Hardman. Lode : une ontologie pour représenter des événements dans le web de données. In *21es Journées Francophones d'Ingénierie des Connaissances (IC 2010), http ://www. ic2010. mines-ales. fr/*, pages 69–80. Ecole des Mines d'Alès, 2010.

Benoît Trouvilliez. *Similarités de données textuelles pour l'apprentissage de textes courts d'opinions et la recherche de produits*. PhD thesis, Artois, 2013.

Willem Robert Van Hage, Véronique Malaisé, Gerben de Vries, Guus Schreiber, and Maarten van Someren. Combining ship trajectories and semantics with the simple event model (sem). In *Proceedings of the 1st ACM international workshop on Events in multimedia*, pages 73–80. ACM, 2009.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

David Vilares, Miguel A Alonso, and Carlos Gómez-Rodríguez. Sentiment analysis on monolingual, multilingual and code-switching twitter corpora. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 2–8, 2015.

Xiaofeng Wang, Matthew S Gerber, and Donald E Brown. Automatic crime prediction using events extracted from twitter posts. In *International conference on social computing, behavioral-cultural modeling, and prediction*, pages 231–238. Springer, 2012.

Yuanyuan Wang, Muhammad Syafiq Mohd Pozi, Goki Yasui, Yukiko Kawai, Kazutoshi Sumiya, and Toyokazu Akiyama. Visualization of spatio-temporal events in geo-tagged social media. In *International Symposium on Web and Wireless Geographical Information Systems*, pages 137–152. Springer, 2017.

Wen-Jiun Wang, Thomas W Haase, and Chia-Hsuan Yang. Warning message elements and retweet counts : An analysis of tweets sent during hurricane irma. *Natural hazards review*, 21(1) :04019014, 2020.

Zhen Wang. *Extraction en langue chinoise d'actions spatiotemporalisées réalisées par des personnes ou des organismes*. PhD thesis, Sorbonne Paris Cité, 2016.

Hong Wei, Hao Zhou, Jagan Sankaranarayanan, Sudipta Sengupta, and Hanan Samet. Delle : Detecting latest local events from geotagged tweets. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Analytics for Local Events and News*, pages 1–10, 2019.

Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 1113–1120, 2009.

Jianshu Weng and Bu-Sung Lee. Event detection in twitter. In *Fifth international AAAI conference on weblogs and social media*, 2011.

Charles A Whitney. The skies of vincent van gogh. *Art History*, 9(3) :351–362, 1986.

Yorick Wilks and Mark Stevenson. The grammar of sense : Is word-sense tagging much more than part-of-speech tagging ? *arXiv preprint cmp-lg/9607028*, 1996.

Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5) :1–46, 2020.

Garrett Wilson and Diane J. Cook. A survey of unsupervised deep domain adaptation. *ACM Trans. Intell. Syst. Technol.*, 11(5), July 2020.

Yiming Yang, Tom Pierce, and Jaime Carbonell. A study of retrospective and on-line event detection. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 28–36, 1998.

Yue Ying, Chen Peng, Chao Dong, Yang Li, and Yan Feng. Inferring event geolocation based on twitter. In *Proceedings of the 10th International Conference on Internet Multimedia Computing and Service*, ICIMCS '18, New York, NY, USA, 2018. Association for Computing Machinery.

Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. Semi-supervised word sense disambiguation with neural models. *arXiv preprint arXiv :1603.07012*, 2016.

Pierre yves Gerlat. Différence entre apprentissage supervisé et apprentissage non supervisé, Jul 2019.

Daniel Zeng, Hsinchun Chen, Robert Lusch, and Shu-Hsing Li. Social media analytics and intelligence. *IEEE Intelligent Systems*, 25(6) :13–16, 2010.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *CoRR*, abs/1707.08114, 2017.

Liang Zhao, Qian Sun, Jieping Ye, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. Multi-task learning for spatio-temporal event forecasting. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1503–1512. ACM, 2015.

Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1) :1–130, 2009.