université
PARIS-SACLAY

Models of diachronic semantic change using word embeddings

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580, Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: mathématiques et informatique
Unité de recherche : Université Paris-Saclay, CNRS, Laboratoire interdisciplinaire des sciences du numérique, 91405, Orsay, France.
Référent: Faculté des sciences d'Orsay

Thèse présentée et soutenue à Paris-Saclay, le 08/02/2021, par

Syrielle Montariol

Composition du jury:

Anne Vilnat Professeur, Université Paris-Saclay	Présidente
Sandra Bringay Professeur, Université de Montpellier	Rapporteur & examinatrice
Antoine Doucet Professeur, La Rochelle Université	Rapporteur & examinateur
Djamé Seddah Maître de Conférence, Université Paris-Sorbonne	Examinateur
Alexandre Allauzen Professeur, Université Paris-Dauphine, ESPCI	Directeur
Jean-Baptiste Janvier Ingénieur, Société Générale	Co-encadrant & examinateur

Thèse de doctorat

NNT: 2021UPASG006

Abstract

Languages are dynamic systems, continuously evolving over time. These changes are carried by the way people use words, and mirror the evolution of various aspects of society such as its technological and cultural environment. We study lexical semantic change: temporal variations in the use and meaning of words, also called *diachrony*.

We explore, compare and evaluate methods to build time-varying embeddings from a corpus in order to analyse word evolution. Recent methods for learning contextualised word embeddings using pre-trained language models such as BERT give new perspectives on this problem. However, they involve a completely new way of training, extracting and using word embeddings. In this thesis, we focus on these tools to tackle the task of semantic change detection, in particular using the BERT model. We propose several approaches to extract and aggregate the contextualised representations of words over time, and quantify their level of semantic change. We focus on the scalability of our approaches, with a view to applying them to large corpora or large vocabulary lists, and on their interpretability, through disambiguating the different usages of a word over time. We evaluate the efficiency of these methods qualitatively and quantitatively, using several annotated corpora. We also apply them in a exploratory fashion to a corpus of COVID19-related newspaper articles and provide interpretations of the detected semantic changes.

Finally, we extend the task of semantic change detection beyond the temporal dimension. First, we adapt it to a bilingual setting: we study the joint evolution of words in two corpora of different languages. Given a word and its equivalent in another language, we propose an experimental framework to characterise how these two words drift in relation to each other. We define a set of bilingual drift scenarios and compare several bilingual diachronic word embeddings systems. Second, we prolong our semantic change detection methods to the synchronic case: detecting semantic variations across different sources or communities on top of time. We apply it to corpora from the financial domain, and compare the semantic shifts with time series of financial indicators. We show that our methods allow to detect and interpret variations in the use of a word across several dimensions, and that it can be linked with real-life events and economic data.

Résumé

Les langues sont des systèmes dynamiques, en constante évolution au cours du temps. Leurs changements reflètent l'évolution de divers aspects de la société tels que l'environnement technologique et culturel. Nous étudions les changements lexico-sémantiques : les variations temporelles dans l'usage et la signification des mots, également appelé *diachronie*.

Nous explorons et évaluons des méthodes de construction de plongements lexicaux variant dans le temps afin d'analyser l'évolution des mots. Les méthodes récentes d'apprentissage de plongements contextualisés à l'aide de modèles de langue pré-entraînés tels que BERT donnent de nouvelles perspectives à ce problème. Cependant, elles impliquent une toute nouvelle façon de former, extraire et utiliser les plongements lexicaux. Dans cette thèse, nous nous concentrons sur ces outils pour aborder la tâche de détection de changements sémantiques, en particulier en utilisant le modèle BERT. Nous proposons plusieurs approches pour extraire et agréger les représentations contextualisées des mots dans le temps, et quantifier leur degré de changement sémantique. Nous nous concentrons sur le passage à l'échelle de nos approches, en vue de les appliquer à de grands corpus ou à de grands vocabulaire, et sur leur interprétabilité, en désambiguïsant les différents usages d'un mot au cours du temps. Nous évaluons l'efficacité de ces méthodes de manière qualitative et quantitative, en utilisant plusieurs corpus annotés. Nous les appliquons également de manière exploratoire à un corpus d'articles de journaux liés au COVID19 et interprétons les changements sémantiques détectés.

Enfin, nous étendons la tâche de détection de changements sémantiques au-delà de la dimension temporelle. Premièrement, nous l'adaptons à un cadre bilingue. Étant donné un mot et son équivalent dans une autre langue, nous proposons un protocole expérimental pour caractériser la façon dont ces deux mots dérivent l'un par rapport à l'autre. Nous définissons un ensemble de scénarios de dérives bilingues des mots et comparons plusieurs systèmes de plongements lexicaux bilingues et diachroniques. Deuxièmement, nous étendons nos méthodes de détection de changements sémantiques au cas synchronique : la détection des variations sémantiques entre différentes sources ou communautés en plus du temps. Nous l'appliquons à des textes du domaine financier et comparons les évolutions sémantiques détectées avec des séries temporelles d'indicateurs financiers, montrant que ces variations peuvent être liées à des événements de la vie réelle et à des données économiques.

Remerciements

Je souhaite remercie mes encadrants Alexandre Allauzen et Jean-Baptiste Janvier, qui se sont bien complétés tout au long de ma thèse. Ainsi que Asanobu Kitamoto, qui m'a encadré lors de mon semestre à Tokyo. Je remercie aussi les membres de mon jury de thèse pour leur bienveillance et leurs conseils avisés.

Côté Société Générale, je remercie Sophie Lavaud, Mamikon Margaryan et Vivien Brunel, sans qui ce projet n'aurait jamais vu le jour; et mes managers Jean-Baptiste Janvier, Flavie Fabre-Azema ainsi que toute l'équipe modélisation.

Merci à toute l'équipe TLP du LISN (ex-LIMSI), en particulier mes co-bureau Djidji, Shu et ma partenaire pour toute cette aventure, Aina.

Pour la rédaction de cette thèse, merci à Antoine Delplace (SG) et Étienne Simon (LIP6); ainsi qu'à Matthieu Labeau et Lauriane Aufrant pour tous leurs conseils et leur soutien.

Merci à mes co-auteurs et collaborateurs qui ont tant contribué à ma thèse: Aina Gari-Soler, Étienne Simon, Corentin Masson, Elaine Zosa, Lidia Pivovrova, Matej Martinc, et Vincent Gouteux.

Enfin, merci à mes colocataires, mes amis et ma famille.

Contents

List of Figures

List of Tables

Introduction

This thesis explores and compare methods to build time-varying word representations from a corpus to analyse their evolution. This is a CIFRE PhD, built as a partnership between the LIMSI-CNRS, University Paris-Saclay, and Société Générale. It is within this context that I chose this research topic, anchoring it into concrete challenges and issues for the company.

Context

As stated by Ferdinand de Saussure : *"Time changes all things: there is no reason why language should escape this universal law."* Languages are dynamic systems, continuously evolving over time. These changes are carried by the way people use words. Most of the time, language users are not aware of this evolution, or do not attempt to control it. These changes stem from various causes; they often mirror evolution of the various aspects of society (Aitchison, 2001) such as the technological and cultural environment. Language contact, through migration of communities or movements of individuals, is also an important factor.

Among all the different aspects of language evolution, we focus on lexical semantic change. Bloomfield (1933) defines it as *"innovations which change the lexical meaning rather than the grammatical function of a form."* An example is the English word "villain", or *vilain* in French. In the 13th century, in both languages, a villain was a village peasant. With the disappearance of the feudal system, the word slowly drifted towards it current meaning, a scoundrel or criminal.

In linguistics, the temporal variations in the use and meaning of a word is called *diachrony*. Detecting and understanding these changes is useful for several sub-fields of linguistic research such as socio-linguistics and historical

linguistics. Words can evolve both in a long-term and in a short-term fashion; the causes and phenomena involved in the evolution differ depending on the granularity. "Villain" was an example of long-term meaning change, spanning several centuries. Short-term changes are most often the consequence of an event that had a resonance in the daily life of the language speakers. Let us take the example of the word "Ukrop" (укроп), meaning "dill" in Russian. Originally a cooking-related word, it has gone through many changes during the Russo-Ukrainian conflict that started in 2014 (Stewart, Arendt, Bell, & Volkova, 2017). At the beginning, Russian people used it as an ethnic slur—a pejorative term—to talk about Ukrainian soldiers. Then, Ukrainian people started to use it to designate their own patriots, as a fusion of "Ukrainian" and "patriot." Thus, in their mouth, it lost its negative connotation. Finally, the political party "Ukrainian association of patriots," abbreviated as "Ukrop," was created June 18th, 2015. Thus, in a few months, the usage of this word went through major changes; moreover, its associated sentiment differs depending on the nationality of the speaker.

This example showcases short-term lexical semantic change, triggered by the war between two countries and fueled by the contact between the speakers in these two countries. We observe synchronic variation on top of semantic change over time: word usage differs between two communities simultaneously. A more common example of word usage variation between communities is the word "wicked": whether it is spoken by an elderly person or a teenager, it will be naturally interpreted as "evil" or "awesome."

Motivation

An immediate motivation of computational methods for language change is to support the work of lexicographers, by analysing large amounts of text with little human power. Various other fields benefit from research in this domain; As stated before, a change in the use of a word is linked with what happens in society. Thus, detecting and analysing word usage evolution can help with the understanding of the associated societal situation. The aforementioned word *ukrop* is a good example, where analysis of word usage change leads to a better understanding of the evolution of the conflict, or helps to track unexpected changes during a crisis. More generally, the detection of semantic change can be used directly for linguistic research or social analysis, by interpreting the

reason of these changes and linking it to real-world events, and by analysing trends, topics and opinions evolution (Gillani & Levy, 2019).

Semantic change can also be used as a support tool for many tasks in Natural Language Processing (NLP). A growing number of historical textual data is digitised and made publicly available nowadays. It can be analysed in parallel with contemporary documents, for tasks ranging from text classification to information retrieval and named entity recognition (Jin, Wi, Kang, & Kim, 2020). Such corpora would be studied with more precision by adding a temporal component to its word representations. In the case of the evolution of the word *ukrop*, without suitable tools and methods to detect the change in its usage, we would not be able to conduct accurate analysis of the text. For example, sentiment analysis performed on text of this period would lead to spurious results if the variations of connotation of this word were not taken into account.

Among other applications, this thesis focus on the financial domain. Financial application of semantic analysis includes all aspects of risk management. Whether it is reputation risk, political risk or systemic risk, the variations of word usage can provide valuable information for analysts.

In order to detect and analyse lexical semantic change, the first step is to capture and represent the meaning of a word. Early work use relative word frequencies and word co-occurrences (Sagi, Kaufmann, & Clark, 2009). Following Bengio, Ducharme, Vincent, and Jauvin (2003) and later Mikolov, Sutskever, Chen, Corrado, and Dean (2013), a line of work proposes to represent a word by a real valued vector: its embedding. This kind of representation takes into account the context in which the word occurs to infer its associated vector. The different word usages observed across the whole corpus are averaged in a unique vector. We also call it *static* embeddings. To study the evolution of word meaning, we need time-varying word embeddings. They rely on the following extension of the distributional hypothesis of Harris (1954): a change in the context of a word mirrors a change in its meaning or usage. This hypothesis leads to the emergence of diachronic word embeddings: word vectors varying over time, following changes in the global context of the word.

More recently, pre-trained language models such as BERT (Devlin, Chang, Lee, & Toutanova, 2019) produce contextualised embeddings: each occurrence of a word is represented by a unique vector. This paves the way for new semantic change detection methods that we investigate in this thesis.

Contributions

In this thesis, we explore, compare and evaluate semantic change detection methods in different settings. Literature on both word representation and semantic change evolved very quickly during the preparation of this PhD. The field of semantic change have seen a sudden acceleration since 2018, with the emergence of dynamic embeddings models, the publication of several literature reviews, the holding of the first ACL Language Change workshop, and the organisation of a SemEval shared task. In parallel, all domains of NLP were shaken by the emergence of pre-trained language models. In the course of this thesis, we went along with this evolution: it results into large disparities between our early and late work. Overall, we focus on embedding-based methods, in particular contextualised embeddings extracted using pre-trained language models.

First, we propose an overview of the literature of diachronic word representations using non-contextual embeddings in Chapter 1. We compared and extended some of these methods in **Learning dynamic word embeddings with drift regularisation**, published at TALN 2019 (Montariol & Allauzen, 2019b).

In Chapter 2, we focus on contextualised embeddings extracted from pre-trained language models. We propose several methods to aggregate these representations at each time slice and to measure semantic change. Some of these methods were published in the paper **Capturing Evolution in Word Usage: Just Add More Clusters?** at the Temporal Web Workshop 2020 (Martinc, Montariol, Zosa, & Pivovarova, 2020a); and the paper **Scalable and Interpretable Semantic Change Detection** at NAACL 2021 (Montariol, Martinc, & Pivovarova, 2021). They were written in collaboration with Matej Martinc, from the Jozen Stefan Institute (Ljubjana, Slovenia) and Elaine Zosa and Lidia Pivovarova, from the University of Helsinki (Finland).

Chapter 3 provides quantitative and qualitative evaluation of the methods introduced in Chapter 2. On top of the ones mentioned before, some results from this chapter come from the paper **Context-sensitive Embeddings not Always Better Than Static for Semantic Change Detection** (Martinc, Montariol, Zosa, & Pivovarova, 2020b) at SemEval 2020 Task 1: Unsupervised Lexical Semantic Change Detection. We also apply our method in an exploratory fashion to a corpus of COVID-related newspaper articles and inter-

pret the detected semantic changes.

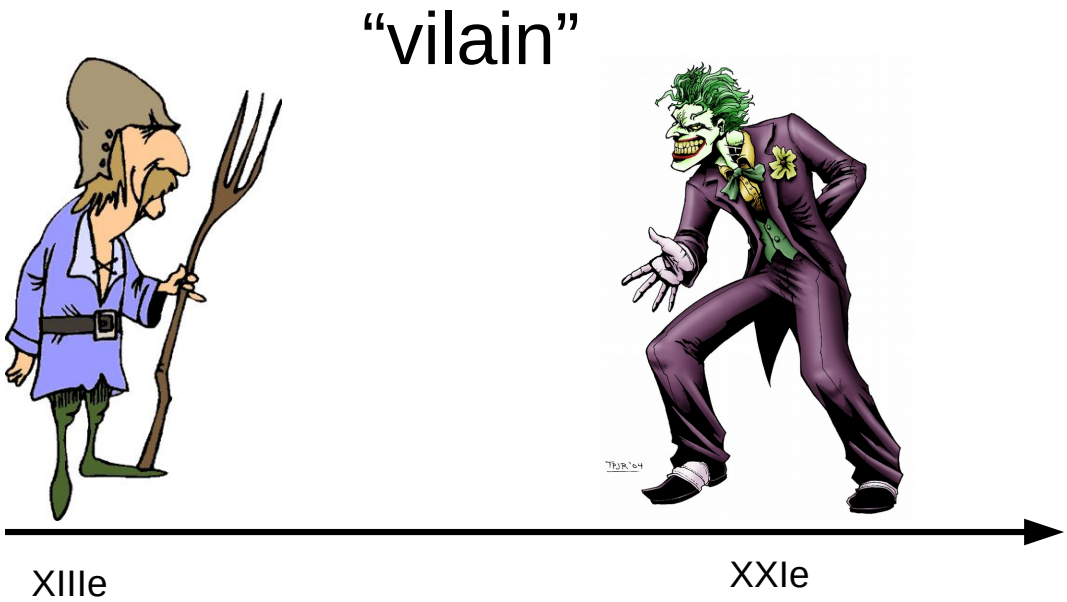
In Chapter 4, we extend the task of semantic change detection to a bilingual setting: we define and motivate the task of bilingual semantic change detection, extend the methods from Chapter 2 to tackle it, and propose an evaluation method relying on the generation of a corpus of synthetic bilingual semantic change. Part of this work was presented at the 2020 DGfS Workshop.

Finally, Chapter 5 focus on financial applications. We prolong our semantic change detection methods to the synchronic case: detecting semantic variation across different sources or communities on top of time. We apply it to corpora from the financial domain, and compare the results with time series of financial indicators. Parts of this work were published in **Studying semantic variations through several dimensions** (Montariol and Allauzen, 2020, TALN 2020) and **Variations in Word Usage for the Financial Domain** (Montariol, Allauzen, and Kitamoto, 2020, Workshop on Financial Technology and Natural Language Processing, FinNLP 2020), in collaboration with Pr. Asanobu Kitamoto from the National Institute of Informatics (Tokyo, Japan).

In the appendices, we describe two of our papers that are related to this work but could not quite fit inside the main body of the thesis.

Appendix C describes the work published under the title **Empirical Study of Diachronic Word Embeddings for Scarce Data** at RANLP 2019 (Montariol & Allauzen, 2019a). It focuses on the problem of data scarcity for diachronic word embeddings, before the rise of contextualised embeddings. We compare three models to learn diachronic word embeddings on scarce data, study the performance of different initialisation schemes and propose a regularisation of these models to better adapt to data scarcity.

Appendix D extends methods from Chapter 2 to fraud detection using contextualised embeddings. We propose a new task of omissions detection in financial reports, along with baselines and evaluation metrics. This work was published at the Workshop on Financial Technology and Natural Language Processing FinNLP 2020 under the title **Detecting Omissions of Risk Factors in Company Annual Reports** (Masson & Montariol, 2020). It was done in collaboration with Corentin Masson from LIMSI.


“vilain”
XIIIe
XXIe

Chapter 1

Diachronic word embeddings: from static to dynamic

Contents

The evolution of word meaning across time has been studied by linguists for a long time. This field evolved quickly with the rise of distributional semantics, starting with corpus linguistic methods and mushrooming towards a very diversified set of methods, from neural word embeddings to topic models.

The first step of any semantic change detection method is to represent a word at different periods in a vector space. In this chapter, we showcase a selection of methods used in the literature to extract time-varying word representations from textual data. These methods can have different focus, from identifying the type of semantic change of a word to quantifying the speed of evolution of its usage.

In Section 1.1, we define semantic change and introduce the different semantic change phenomena tackled in the literature. Then, we present two types of methods that associate a unique representation to a word in a time slice: frequency-based representation methods (Section 1.2), and methods based on neural word embeddings. Before presenting the latter in sections 1.4 and 1.5, we give a short summary of the classical word embeddings methods (Section 1.3.1). In Section 1.6, we expose methods to detect semantic change and interpret it using these diachronic embeddings. Finally, in contrast to the other methods presented in this chapter, Section 1.7 summarises methods that disambiguate the various senses of a word across time, before the rise of pre-trained language model for contextualised embeddings extraction.

1.1 Semantic change: introduction

The evolution of word meaning across time is an old research topic in linguistics (Bréal, 1899; Stern, 1931; Bloomfield, 1933). With the expansion of this research field towards computational linguistics, its denomination is becoming more and more diverse. It can be called many different ways, depending of the community tackling it or the period of the study: semantic change, language change, diachronic shift, semantic drift, diachronic conceptual change, and nearly all possible combinations of these words. In this thesis, we mainly use the term "semantic change."

1.1.1 Definition and categorization

As explained in the introduction, semantic change is the adjustment of word meaning to the evolution of all aspects of the speaker's environment (societal, cultural, technological...) (Blank & Koch, 1999). Lexical semantic change as the evolution of word meaning can belong to various non-mutually-exclusive categories. In practice, in the literature of computational semantic change detection, the types of change that are investigated can be broadly defined as follow (Tahmasebi, Borin, & Jatowt, 2018):

- **Novel word**: a new word with a new sense. For example, the word *selfie* is the slang term for "self-portrait." Its first known use is in an Australian internet forum in 2002; then, it promptly spread to the whole word, and

was even elected as the "Word of the Year" in 2013 by the Oxford English Dictionary.

- **Novel word sense**: a novel word sense (related or not to an existing sense) attached to an existing word. For example, the word *apple*, originally only referring to the fruit, started a few decades ago to be associated with the computer brand "Apple." This evolution is illustrated in Figure 1.6.
- **Broadening** and **Narrowing**: a word sense becoming broader or narrower in meaning. For example, the Old English word *hund* meant "dog," while in Modern English *hound* refers to a specific breed of dogs used in fox-hunting.
- **Join** and **Split**: two word senses that exist individually and then join, or a word sense that splits into two individual senses. For example, the word *mantra* could be associated with both an action—chanting and prayer—and its effect—salutations, benedictions—at the beginning of the 20th century. These two usages have merged into one today, where the word is used as "blessings"or "spell."
- **Pejoration** and **Amelioration**: the connotation of the word becomes more negative or more positive. For example, the English word *dude* used to mean "fastidious man" when it appeared at the end of the 19th century; today, it only means "man" with a neutral connotation.
- **Death**: a word sense that is no longer used.
- **Change**: any significant change in sense that subsumes all previous categories.

Taking back the example of the Russian word *ukrop*, meaning "dill": the **new sense** of "Ukrainian soldier" is added with pejorative connotation, before going through **amelioration** and **broadening** towards "Ukrainian patriot" in the mouth of Ukrainian people, and finally gaining the **new sense** of the political party in 2015.

However, the idea behind the sense of a word is itself a controversial subject. Senses are defined from a lexicographic point of view, as listed in a dictionary. Two situations can be distinguished:

- **Polysemy** refers to a word that can be used to express several different meanings, no matter how big the difference between the meanings. For example, in the two following sentences, the word *newspaper* refers both to the object and the company:
  (a) The *newspaper* got wet in the rain.
  (b) The *newspaper* fired some of its editing staff.
- **Homonymy** refers to several words having the same spelling but unrelated meanings. For example, in the sentences below, *bank* is spelled the same but has completely unrelated senses:
  (a) I sat on the river *bank*.

(b) I opened a *bank* account.

In the two examples, the inclusion of the word in the polysemy or homonymy category is obvious. However, often, the difference is not clear, as the actual distinction between the two categories is gradual. Similarly, the difference between *polysemy*—several senses—and *identity*—a unique sense—is not always obvious. For example, a sense of the word *phone* is "a system that uses wires and radio signals to send people's voices over long distances." However, the first phones from the 19th century have little in common with today's devices, in operation, appearance and in use. When should we consider the sense to be different?

Kilgarriff (1997) argues that the concept of word sense cannot be "a workable basic unit of meaning," as the discreteness behind this concept is greatly limiting. The emergence of a new sense is progressive. The usage of a word becomes more diversified over time, until it is decided that we face two distinct senses. For example, before the COVID19 pandemic, *sheltering* referred to seeking safety during a circumscribed event, usually bad weather. Nowadays, it is used to refer to a prolonged period of social isolation: the meaning evolved with the crisis situation.

Considering this ambiguity behind word senses, we study semantic change as any variation of word usage revealed by a change in its context; even if it does not involve a change in sense from the lexicographic point of view.

1.1.2 Tasks and methods

In this work, we consider the problem of semantic change detection on the following basis: we have a corpus of documents, each associated with a date. We divide it into several time periods, according to the chosen granularity. A time slice can be of any order of magnitude, from days to centuries, depending on the corpus and the phenomenon under study. This leads us to a set of pairs (text, time period). For a given target word, we extract a signal from all the contexts in which the word is used in each time period. This signal is a representation of the semantic information of the word at each period. We study semantic change as the evolution of this representation across time. All models presented in the following chapters aim at extracting this signal.

Many tasks are involved in this problem. The main computational linguistics tasks derived from this phenomenon are:

- Quantifying the degree of semantic change of each word in a corpus (ranking task)
- Detecting whether words undergo semantic change or not in a corpus (binary task)
- Detecting when semantic change happens.

- Interpreting the change undergone by a word.

Other tasks can involve studying semantic change in relationship with other factors such as the frequency of words and their level of polysemy (Hamilton, Leskovec, & Jurafsky, 2016). It can also deal with characterizing other aspects of semantic change: change in the connotation of a word (Jatowt & Duh, 2014), or change restricted to a specific domain or community. For example, Kutuzov, Velldal, and Øvrelid (2017) use diachronic embeddings to study the evolution of semantic relations between pairs of words and apply it to predicting insurgent armed groups based on geographical locations. In the psychology domain, Vylomova, Murphy, and Haslam (2019) analyse semantic changes of harm-related concepts such as "harassment" or "addiction" and highlight a phenomenon of broadening of these concepts through time.

We give an overview of methods to build time-varying word representations. The systems presented here, as well as most computational linguistics methods, rely on the distributional hypothesis (Harris, 1954). The heart of this hypothesis can be summarised this way: semantically similar words tend to appear in similar linguistic contexts. This has immediate application for extracting semantic properties of a word using corpus statistics.

Historically, the first approaches to diachronic modeling were based on relative word frequencies and distributional similarities (Hilpert, 2006). The use of systems based on word embeddings is more recent and has undergone a surge of interest these last three years with the publication of several articles dedicated to a literature review of the field (Kutuzov, Øvrelid, Szymanski, & Velldal, 2018; Tahmasebi et al., 2018; Tang, 2018).

In sections 1.2 to 1.5, we describe the methods that do not take the polysemy of words into account: they collapse all possible usages of a word during a period into a single representation. All these methods divide a corpus into time slices and extract word representations for each time slice. Section 1.7 describe methods that disambiguate the senses of the words in each time slice to track their evolution.

1.2 Corpus linguistics and semantic change

The study of lexical semantic change is a traditional topic in linguistics, involving a lot of manual work: analyses were usually done at a very small scale. With the growing amount of digitised texts spanning a large historical period, the fields of computational linguistics and computer sciences started taking interests in diachrony during the last decade. Rather than actually studying lexical semantic change, most papers in these fields propose methods and models to detect it in an automatic and reliable way.

In this section, we compile a few semantic change detection methods relying on word frequencies and word co-occurrences.

Word frequency. Originally, methods to study diachrony on temporal corpora made use of the distributional statistics of the words over time. They start with computing word frequencies for each time period. For example, Michel et al. (2011) compute word frequencies on the Google Books corpus and analyse their evolution as the reflection of the linguistic and cultural phenomena that occurred between the 19th and the 21st centuries. This allows to observe the evolution of grammar, the appearance and disappearance of trends, and even to detect censorship. It can also be applied to semantic change detection; indeed, it can be expected that if a word becomes more popular (thus having its frequency increased), it might have either changed its meaning or gained a new one. As a measure of semantic change, one can for example use the ratio between frequency of word occurrence at two successive time slices (Gulordava & Baroni, 2011).

Word co-occurrences. Word collocations are a valuable method to measure and describe linguistic phenomena such as semantic relationship (Church & Hanks, 1989; Daille, 1994). Thus, temporal analysis of word collocations can be used as a tool to study semantic change.

Hilpert (2006) study the relationship between two words across several time periods. To model this relationship, they compute the statistical dependency between pairs of words at different time slices in a corpus, and compare them in order to measure semantic change. More precisely, a pair of words is said to be repealing or attracting over time, depending on the value of the binomial p-value of the pair—to compute their expected co-occurrence frequency—compared to their actual number of co-occurrences. Semantic change through statistical dependency can be observed for example in the word *apple*: its relationship with the word *phone* is very low is the 1980s and rises quickly in the 2000s.

Sagi et al. (2009) represent words by applying Singular Value Decomposition on a reduced version of the co-occurrences matrix extracted from a temporal corpus, leading to 100-dimensional vector representations for each word. From this, they create context vectors of each occurrence of a given target word by taking word representations of all surrounding words in a window of size 15. The semantic density of the target word at a given time slice is computed by taking the average pairwise cosine similarity of the set of context vectors of this time slice. This measure allows the authors to identify changes such as narrowing and broadening of senses over time, respectively indicated by an increasing semantic density or a decreasing one. For example, they apply their method to a corpus of texts spanning the periods of Middle English (1150–1500A.D.) and Early Modern English (1500–1710A.D.). They observe the broadening of the set of senses of the word *do* during these periods, as part of the process of grammaticalization[1]. Indeed, in Old English, *do* was

[1]When a content word becomes a function word. Content words usually have a smaller diversity of contexts compared to function words.

used as a verb with a causative and habitual sense (as in *"Do you harm?"*), while in more modern English it took on a functional role as in *"Do you want tea?"*). As a result, the authors observe a substantial diversification in context between the two periods under study. However, this methods does not discriminate between senses; it only allows to quantify the degree of semantic change.

Similarly, the distributional semantics model of Gulordava and Baroni (2011) uses co-occurrences matrices to compute Local Mutual Information (LMI) scores. They quantify the similarity between two words by how frequently they appear within the same context in a corpus, with the context being defined as a window of n words around the target word. They choose a window of 2 words, meaning that only 2-grams are considered to build the co-occurrence matrix. Given two words w_1 and w_2, let us denote $C(w_1)$ the number of occurrences of the word w_1 and $C(w_1, w_2)$ the number of co-occurrences of w_1 and w_2 in the same window. From this, the LMI between two words that co-occur is computed as follow (N is the total number of 2-grams in the corpus):

$$\text{LMI}(w_1, w_2) = C(w_1, w_2) \log_2 \frac{C(w_1, w_2) \times N}{C(w_1)C(w_2)} \tag{1.1}$$

Note that the LMI is derived from the more broadly used Pointwise Mutual Information (PMI):

$$\text{LMI}(w_1, w_2) = \text{PMI}(w_1, w_2)C(w_1, w_2)$$

Defining the vector representation $\mathbf{v}_i$ of the context of a word w_i as the set of LMI between w_i and all the words in the vocabulary, the similarity between two words w_1 and w_2 can be defined as the cosine similarity between their context vectors: $\text{sim}(w_1, w_2) = \cos(\mathbf{v}_1, \mathbf{v}_2)$. By building a new co-occurrence matrix at each time period, the authors can compute the similarity between a word at one time period and the same word at another period. They apply this methods to two decades of the large Google Ngrams Corpus (see Section 3.1.1 for more details), comparing word contexts between the 1960s and the 1990s. They identify two types of phenomena. First, semantic change in its linguistic definition—one of the categories listed in Section 1.1, such as emergence of a new sense—can be seen through examples such as *disk* that gained a technological meaning. Second, they detect change in the main context in which the word is used, without the word going through an actual semantic change. For example, they capture the evolution of the usage of the word *parent*, often appearing in collocation with the word *single* in the 1990s, which was not the case in the 1960s.

1.3 Neural word embeddings and diachronic methods

Following Bengio et al. (2003), a line of work proposes to represent a word and its usage by a learned real valued vector: its word embedding. This kind of representation takes into account the word in its observed contexts to infer a vector representation. It is the direct evolution of the distributional similarity methods used in the previous section. Several methods to train word embeddings appeared successively. They all rely on the hypothesis that words with similar meanings appear in similar contexts (Harris, 1954). First, Mikolov, Chen, Corrado, and Dean (2013) proposed the Word2Vec framework with two algorithms, Continuous Bag of Words (CBOW) and Skip-Gram. Levy and Goldberg (2014) demonstrate that Word2Vec's Skip-Gram implicitly factorizes a word-context PMI matrix. One year later, Pennington, Socher, and Manning (2014) propose Glove, a system that relies on the factorisation of word-context co-occurrence matrix to build embeddings. Instead of taking raw co-occurrence probabilities to encode the meaning of words, they use the ratio of the co-occurrence probabilities of three words. Then, Bojanowski, Grave, Joulin, and Mikolov (2017) release FastText, an algorithm that overcomes the issue of out-of-vocabulary words by taking subwords as smallest component to build embeddings instead of full words.

To give a general idea of the principle of word embeddings and build on it in the following sections, we focus on the most famous one, the Word2Vec system described by Mikolov, Chen, et al. (2013). Then, we introduce the possible strategies for a diachronic extension of neural word embeddings.

1.3.1 Focus on Word2Vec embeddings

The Word2Vec framework consists of two models, compared in Figure 1.1: Continuous Bag of Words (CBOW) and Skip-Gram. Both are two-layer neural networks aiming at learning linguistic contexts of words and reconstructing it into a vector space. CBOW predicts which word is most likely to appear given its context, while Skip-Gram uses the network to predict the context words around a given target word. Thus, CBOW treats the full context of a word as one observation; on the contrary, Skip-Gram considers each context-target pair as a new observation. This makes in general Skip-Gram more appropriate for large corpora and CBOW to small ones.

Let us dive a bit deeper into the Skip-Gram model. We consider a corpus written at a given period t. A word i is represented by its vectors $u_i^{(t)}$ and $v_i^{(t)}$ at time t, with $u_i^{(t)}$ being the representation of the word i as a central word and $v_i^{(t)}$ its representation as a context word. The matrices $U^{(t)}$ and $V^{(t)}$ gather the embedding vectors for the whole vocabulary at time t. We use notations indexed by t to enable immediate adaptation for the notations of the following

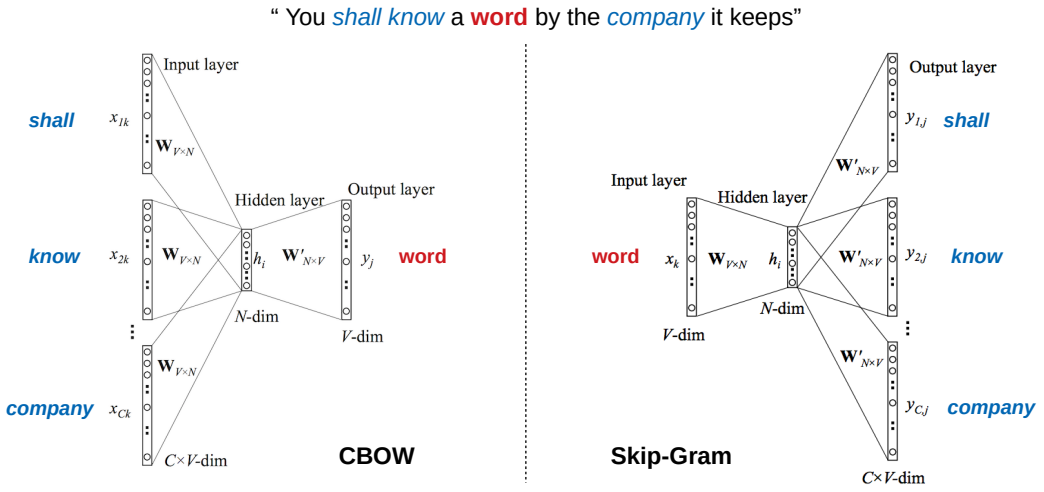


Figure 1.1 – Comparison of model architecture for Skip-Gram and CBOW (Figure derived from Rong (2014)).

sections.

The objective is to maximise the likelihood of the observed corpus, that is the probability of each word j to occur in the context of the word i in the dataset. The probability of a word i to appear in the context of a word j is modeled by $\sigma(u_i^{(t)\intercal} v_j^{(t)})$, where σ is the sigmoid function. The context is made of a fixed number of surrounding words and each word in the context are considered independent of each other given the target word.

The Skip-Gram model can be trained using hierarchical softmax or negative sampling; the latter is the most frequent, leading to the use of the common abbreviation SGNS (Skip-Gram with Negative Sampling, Mikolov, Sutskever, et al., 2013). It avoids to update the full vocabulary, sampling only a pre-defined number of negative examples from a noise distribution. Given the independence assumption, the negative sampling strategy associates to each positive example—an observed word-context pair (i, j)—a set of negative examples, which are words that do not appear in the context of the central word.

Let L be the size of the vocabulary and n_{ijt}^{+} denote for the time period t the total number of positive examples for the pair (i, j), n_{ijt}^{-} the number of negative examples, and their sum $n_t^{\pm}$. The objective function can be defined as the following log-likelihood:

$$\log p(n_t^{\pm} | U^{(t)}, V^{(t)}) = \mathcal{L}_{pos}(U^{(t)}, V^{(t)}) + \mathcal{L}_{neg}(U^{(t)}, V^{(t)}) \tag{1.2}$$

$$= \sum_{i,j=1}^{L} (n_{ijt}^{+} \log \sigma(u_i^{(t)\intercal} v_j^{(t)}) + n_{ijt}^{-} \log \sigma(-u_i^{(t)\intercal} v_j^{(t)})) \tag{1.3}$$

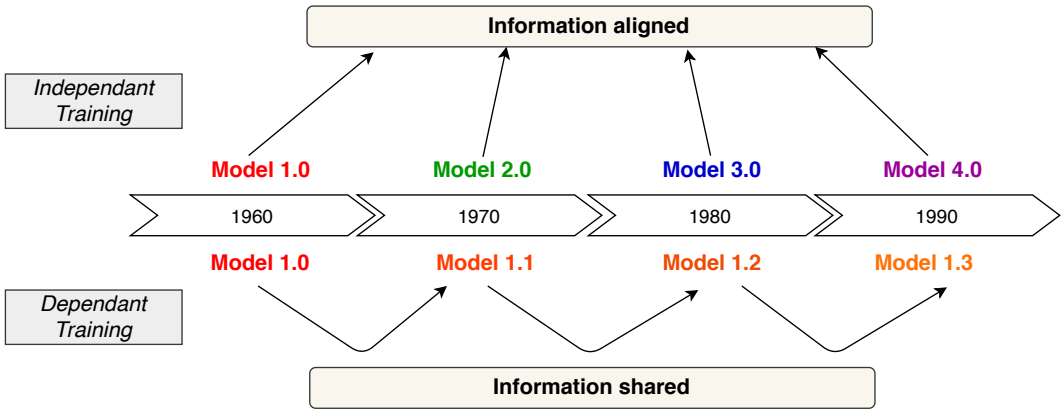


Figure 1.2 – Comparison of the two types of training methods on a corpus ranging between the 1960s and the 1990s, divided into decades.

1.3.2 Classification of diachronic embeddings methods

Neural word embeddings average in one vector the different word's usages observed across the whole corpus. This *static* representation hypothesis turns out to be limited in the case of temporal datasets. The diachronic counterpart of this hypothesis is that a change in the context of a word mirrors a change in its meaning or usage. Thus, one can train word embeddings that evolve across time, following the changes of their usage in a corpus. Several methods exist to train such embeddings. They can be broadly divided into *independent* learning of embeddings for each time slice, and *dependant* learning strategies that use information from the whole period under study. Figure 1.2 summarises the two systems.

For independent methods, static embeddings are trained independently on each time slice of the corpus. Then, they are compared using solutions like alignment or anchoring (Section 1.4).

The second set of methods involve either incremental training or joint training (Section 1.5). For incremental training, we learn the embeddings in a Markovian fashion: the models are fine-tuned from each time slice to the next one by using the embeddings of the previous time slice as initialisation. A dynamic extension to this system is to add a regularisation term during training in order to control the drift of the embeddings at each new time slice. Finally, joint training allows to learn all the embeddings of a word for the different time slices jointly on the full corpus.

The key difference between these two types of methods is how the time information is being exploited: as a categorical nominal variable, in the independent case, or as an ordinal or continuous variable, in the dependant case.

1.4 Independent methods

Word embeddings are trained on each time slice of a temporal dataset *independently*. Due to the stochastic aspect of word embeddings training processes, the vector spaces for each time slice are different, making them not directly comparable. Several methods can be implemented to tackle this issue.

This problem did not appear in earlier work. Indeed, the count-based methods presented in the previous section rely on a co-occurrence matrix. Its columns correspond to words that usually occur in most time slices; thus, the intersection of the co-occurrence matrices of all time slices can be used for comparison.

1.4.1 Alignment

Kulkarni, Al-Rfou, Perozzi, and Skiena (2015) align the embeddings spaces of all time slices into a common vector space by learning a linear mapping between the word representations of the nearest neighbours of a word at the different time periods. Indeed, while training a model twice on the same data leads to different vectors for a given word, its nearest neighboring words will be the same. Thus, the authors define as $\text{knn}(u_i^{(t_1)})$ the set of k-nearest neighbors of the representation $u_i^{(t_1)}$ of word i at time t_1. The goal is to find the best linear transformation W_{t_1,t_2} to map a word vector from the embedding space of the source period to the one of the target period. In order to do so, they solve the following optimization problem:

$$W(u_i)_{t_1,t_2} = \arg\min_W \sum_{u_j^{(t_1)} \in \text{knn}(u_i^{(t_1)})} \|u_j^{(t_1)} W - u_j^{(t_2)}\|^2 \tag{1.4}$$

Similarly, Hamilton et al. (2016) optimise a geometric transformation to find the best alignment. More precisely, they use the orthogonal Procrustes (Schönemann, 1966) to align embeddings trained using the Skip-Gram with negative sampling model (SGNS, Mikolov, Sutskever, et al., 2013).

We note $U^{(t)}$ the word embedding matrix at time t. Before making the alignment, the embedding matrices are often mean-centered and normalised by the number of different words in the time slice. Following thorough experiments, Schlechtweg, Hätty, Del Tredici, and Schulte im Walde (2019) underlines that mean-centering as a preprocessing step for the embedding matrices before alignment is crucial for the efficiency of the method. Orthogonal Procrutes consists in finding the mapping W_{t_1,t_2} between the two embedding spaces $U^{(t_1)}$ and $U^{(t_2)}$ which minimizes the sum of squared Euclidean distances between the image of the source embedding space $U^{(t_1)}W$ and the target em-

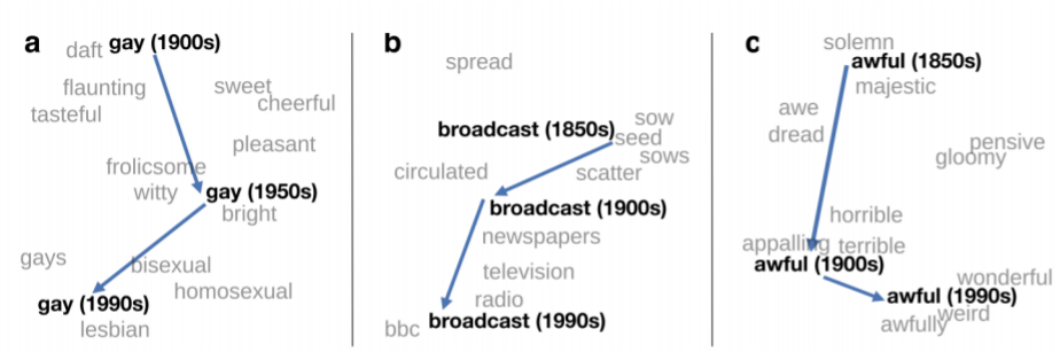


Figure 1.3 – Two-dimensional visualization of semantic change in English using SGNS vectors (Hamilton, Leskovec, & Jurafsky, 2016).

bedding space $U^{(t_2)}$:

$$W_{t_1,t_2} = \arg\min_{W^\intercal W=I} \sum_i \sum_j \|U_i^{(t_1)} W - U_j^{(t_2)}\|^2 \tag{1.5}$$

The target embedding space is usually chosen as the one from the last time period, as the most recent time slices have usually the most data in historical corpora. For this task, the authors use an orthogonality constraint for W. Thus, equation 1.5 is equivalent to maximizing a dot product: the optimal solution W can be written $AB^\intercal$ where $A\Sigma B^\intercal$ is the Singular Value Decomposition of ${U^{(t_2)}}^\intercal U^{(t_1)}$.

Figure 1.3 shows a few examples detected by Hamilton et al. (2016) using alignment of SGNS vector spaces. The word *gay* shifted from the neighboring of "cheerful" and "flaunting" in the 1900s to its current meaning *homosexuality* (part **a**). The word *broadcast* originally meant "spreading seeds" during the 19th century; it drifted towards its current meaning "transmitting a signal" with the rise of television and radio in the 20th century (part **b**). Finally, the adjective *awful* drifted from the positive connotation "solemn" and "majestic" towards the negative connotation "terrible" and "weird." All these shifts are validated by linguists in the diachrony literature.

Several variants of this method appear in the literature, even though the orthogonal Procrustes used by Hamilton et al. (2016) remains the most popular (Shoemark, Liza, Nguyen, Hale, & McGillivray, 2019).

First, Zhang, Jatowt, Bhowmick, and Tanaka (2015) use vector spaces alignment to find temporal counterparts of words. For example, *iPod* today is the temporal counterpart of *walkman* 20 years ago. For this purpose, they use a set of temporal pairs of words as anchors for the alignment instead of using the full vocabulary like Hamilton et al. (2016). Thus, the transformation matrix W_{t_1,t_2} is obtained by minimizing the sum of squared Euclidean distances between transformed anchor words and their equivalent at the previous time slice. The temporal pairs to be used as anchors must have sufficient frequency in the two time periods. They rely on the assumption that very frequent

terms (e.g., woman, water, dog, see, three) are semantically stable across time, especially across a short period of time. They also add a regularization term on matrix W to equation 1.5 to overcome the problem of over-fitting. Azarbonyad et al. (2017) also use the most frequent words to conduct the alignment. They add a set of stopwords to the list of anchor words.

Similarly, Tsakalidis, Bazzi, Cucuringu, Basile, and McGillivray (2019) argue that aligning the vectors of the whole vocabulary is not appropriate for semantic change detection, as it tends to lower the disparities between the different vector spaces. In order to find the most stable words, they first conduct alignment on the full vocabulary; then, they measure the displacement error of all words between the two time slices using cosine distance. The most stables words are selected as anchor. This setup leads to different sets of anchor words being selected at each time step: thus, they average the displacement error across all time slices to select stable words for the full corpus. Then, they compute the transformation matrix for the alignment using only the selected anchor words, and apply this transformation to the full vocabulary.

Finally, Lin, Wan, and Guo (2019) also use stable words to improve the alignment of vector spaces between time slices; however, instead of training the embeddings for each time slice separately before aligning, they do both steps jointly during training. Concretely, they iteratively extract stable words from the corpus and use them as anchors to build diachronic constraints for the alignment. They argue that stable words can differ across time; for example, the context of the word *president* is relatively stable between periods of election and vary a lot right before and after.

1.4.2 Temporal Referencing

An alternative method to alignment for semantic shift detection with static word embeddings is to treat the same words in different time periods as different tokens in order to get time-specific word representations for each period. Thus, only one embedding model needs to be trained and no aligning is needed.

For this purpose, Dubossarsky, Hengchen, Tahmasebi, and Schlechtweg (2019) use Temporal Referencing (TR) with the SGNS model. This method is inspired by Ferrari, Donati, and Gnesi (2017) who use it to compute similarity between technical terms in different domains. Each word w of the list appearing at time t is replaced in the corpus with its time-specific equivalent w_t before training semantic representations on the full corpus. Thus, each time-specific token has a different representation while the context words stay stable through time. This is a common hypothesis in the literature of diachrony (Rudolph & Blei, 2018). TR is more robust to noise and less computationally demanding than the alignment methods, as it allows to analyse the evolution of a list of target words in a corpus covering a long period without training embeddings at each time slice. However, it can only be performed over a set of pre-defined target words.

Huang and Paul (2019) extend this method to FastText embeddings (Bojanowski et al., 2017). They consider all words of the vocabulary as targets for the Temporal Referencing method, replacing them by their temporal counterpart *w_t* in each time slice. As FastText embeddings are trained using a sub-word sharing framework, the model learns time-independent representations of each word despite the temporal suffix added to all words. The authors illustrate this idea with the word *where*. It is replaced by *where1* in the first time slice. FastText ecodes this word with 3-grams of characters the following way:

$$< \text{wh, whe, her, ere, re1, e1} >, < \text{where1} >$$

Thus, the temporal identity of the word is retained while the same words at different time slices will have similar representation, as they share the same morphological forms.

1.4.3 Anchor methods

Azarbonyad et al. (2017) use a simple method to compare vectors in two different representation spaces. They rely on graph theory, where the similarity of two nodes can be computed using the similarity of their neighbors. Considering each embedding space as a graph and each word as a node, they compute the cosine similarity of a word embedding with the embeddings of its neighbors. A word is considered stable if it has the same neighbors in both embeddings spaces, and the level of stability can be computed in an iterative way, by taking into account the closest neighbors of a word, the neighbors of its neighbors... They show that this stability measure can be combined with the drift measure obtained through the alignment method, to increase the accuracy of semantic change detection.

Eger and Mehler (2016) and Yin, Sachidananda, and Prabhakar (2018) compare similarity vectors: they create "second-order embeddings" by representing each word by its similarity with all other words in the vocabulary. This leads to high-dimensional embeddings that can be compared between time slices without requiring any alignment. The anchors—words with which the similarity is computed inside each representation space—can be selected in a specific domain to reflect shifts along a specific direction instead of taking the whole vocabulary as a basis. To measure the distance between two words, Eger and Mehler (2016) use the cosine similarity while Yin et al. (2018) use the inner product.

More recently, Gonen, Jawahar, Seddah, and Goldberg (2020) propose a method relying on intersection of nearest neighbors, similar to Azarbonyad et al. (2017). Given the hypothesis that words that undergo semantic change see their nearest neighbors changing in the representation space, each word is represented by the set of its top-k nearest neighbors according to inner product.

The highest the size of the intersection of the nearest neighbors between two vector spaces, the more stable the word is. To increase the stability of the method, they choose a large number of 1000 neighbors per word.

1.5 Joint and incremental methods

The alignment, temporal referencing and anchoring methods presented in the previous section treat all time slices independently. It means that they do not explicitly model the ordering of time information in the corpus. The methods presented in this section all consider time as ordinal information instead of nominal. Time slices do not necessarily have to be of equal size, but their order is taken into account when training; either incrementally (using the previous time slice as information for the next one) or jointly (using all time slices together).

1.5.1 Incremental fine-tuning

One of the first works that used word embeddings for semantic change detection was conducted by Kim, Chiu, Hanaki, Hegde, and Petrov (2014), who leverage the SGNS algorithm. They propose the incremental model fine-tuning approach, where the weight of the model, trained on a given time period, are used to initialize the weights of a model trained on the following time period.

The SGNS model for a unique period was exposed in section 1.3.1. Considering a corpus divided into T time slices, the incremental version works as follows:

1. For the first time slice, the embeddings matrices $U^{(1)}$ and $V^{(1)}$ are initialised using a Gaussian random noise $\mathcal{N}(0, 1)$ and trained according to equation 1.2.

2. Each $V^{(t)}$, $U^{(t)}$ are initialised with values of the previous time slice $V^{(t-1)}, U^{(t-1)}$ and trained incrementally.

This way, the word vectors of each time step are all in the same vector space and directly comparable. The main idea behind this methods is that when a word is used in similar contexts in two distinct time slices, its vector will be updated only slightly during fine-tuning. If the context varies a lot, the update of its vector would be more important.

Some improvements of the approach were later proposed by Peng, Li, Song, and Liu (2017), who use hierarchical softmax instead of negative sampling, and by Kaji and Kobayashi (2017), who improve the negative sampling estimation.

1.5.2 Dynamic Word Embeddings (DWE)

This second method relies on the Bayesian extension of the SGNS model described by Barkan (2017). Its goal is to control the drift of the embeddings in order to enforce smoother trajectories than the classical incremental method decribed above. The main idea is to share information from one time slice to another, allowing the embeddings to drift under the control of a diffusion process. A full description of this approach, denoted as the filtering model, can be found in Bamler and Mandt (2017).

In this model, the vectors $u_i^{(t)}$ and $v_i^{(t)}$ are considered as latent probabilistic vectors. Under a Gaussian assumption, they are represented by their means $(\mu_{u_i^{(t)}}, \mu_{v_i^{(t)}})$ and covariance matrices $(\Sigma_{u_i^{(t)}}, \Sigma_{v_i^{(t)}})$. The covariance matrices are restricted to be diagonal. They are initialised for the first time slice with respectively a zero mean vector and an identity covariance matrix.

Similarly to the incremental updating method, as many copies of the bayesian skip-gram model as time slices are successively trained. The temporal drift from one time step to another follows a Gaussian diffusion process with zero mean and variance D. This variance is called the *diffusion* constant and has to be tuned along with the other hyperparameters. Moreover, at each time step a second Gaussian process with zero mean and variance D_0 is added to prevents the embedding vectors from becoming too large. It results in the following distributions over the embeddings matrices $U^{(t)}$:

$$\begin{aligned} U^{(1)} &\sim \mathcal{N}(0, D_0) \\ p(U^{(t)}|U^{(t-1)}) &\propto \mathcal{N}(U^{(t-1)}, D)\ \mathcal{N}(0, D_0). \end{aligned} \tag{1.6}$$

The same equations stand for $V^{(t)}$. Training this model requires to estimate the posterior distributions over $U^{(t)}$ and $V^{(t)}$ given $n_t^{\pm}$. This (bayesian) inference step is unfortunately intractable. In Bamler and Mandt (2017), the authors propose to use variational inference (Jordan, Ghahramani, & et al., 1999) in its online extension (Blei, Kucukelbir, & McAuliffe, 2017) to provide an approximation of the posterior distributions. The principle of variational inference is to approximate the posterior distribution with a simpler variational distribution $q_\lambda(U, V)$ where λ gathers all the parameters of q. This variational posterior will be iteratively updated at each time step. To approximate it, we have to minimise the Kullback-Leibler (KL) divergence from the posterior p to the variational distribution q. This is equivalent to maximising the evidence lower bound (ELBO, Kingma and Welling, 2014). The final objective to be maximised can be written as follows:

$$\begin{aligned} \mathcal{L}_t(\lambda) = &\mathbb{E}_{q_\lambda}[\log p(n_t^{\pm}|U^{(t)}, V^{(t)})] \\ &+\mathbb{E}_{q_\lambda}[\log p(U^{(t)}, V^{(t)}|n_{1:t-1}^{\pm})] \\ &-\mathbb{E}_{q_\lambda}[\log q_\lambda(U^{(t)}, V^{(t)})]. \end{aligned} \tag{1.7}$$

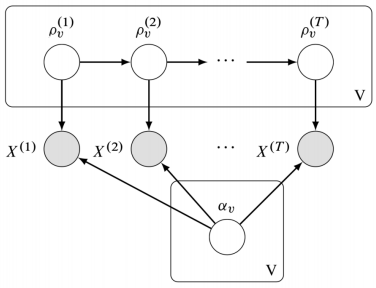


Figure 1.4 – Scheme of DBE system for text data $X^{(1)}, \dots X^{(T)}$ in T time slices (Rudolph & Blei, 2018).

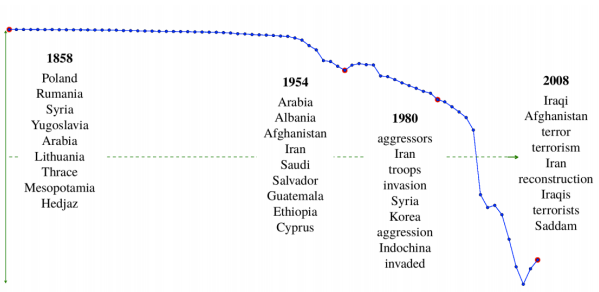


Figure 1.5 – One-dimensional projection of the embeddings of the word *iraq* between 1858 and 2009 using PCA, with nearest embeddings for selected points (Rudolph & Blei, 2018).

This function is the sum of three terms: the log-likelihood (computed following equation 1.2), the log-prior (which enforces the smooth drift of embedding vectors, sharing information with the previous time step), and the entropy term (which can be computed analytically).

1.5.3 The Dynamic Bernouilli Embeddings Model (DBE)

The Dynamic Bernouilli Embeddings rely on a system similar to the Dynamic Word Embeddings from Bamler and Mandt (2017). The main difference lies in the embedding model; instead of the Skip-Gram, they use the *Exponential Family Embeddings* (EFE, Rudolph, Ruiz, Mandt, and Blei, 2016), a probabilistic generalisation of the *Continuous Bag-of-Words* (CBOW) model of Mikolov, Sutskever, et al. (2013). Moreover, the approximate learning criterion relies on negative sampling instead of the bayesian inference of the DWE model.

We give an overview of the EFE below before describing the dynamic version called *Dynamic Bernoulli Embeddings* (DBE) proposed by Rudolph and Blei (2018).

Bernoulli embeddings. Let us consider a vocabulary of size V. Each word v at position i in the corpus is associated with a one-hot vector $\mathbf{x}_i \in \{0,1\}^V$ where only the v-th element is 1. Its associated context $\mathbf{x}_{\boldsymbol{c}_i}$ gathers the embeddings of all the words in the window $\boldsymbol{c}_i$ of size C around i.

The model predicts the central word vector $\mathbf{x}_i$ conditionally to its context vectors $\mathbf{x}_{\boldsymbol{c}_i}$ following a Bernoulli distribution for each possible word: $\mathbf{x}_{iv}|\mathbf{x}_{\boldsymbol{c}_i} \sim \text{Bern}(p_{iv})$. The natural parameter of the Bernoulli distribution p_{iv} is computed using the embedding vector $\boldsymbol{\rho}_v$ and the context vectors $\boldsymbol{\alpha}_{v'}$—

equivalent to u and v in the Word2Vec framework—as follows:

$$p_{iv} = \sigma\left(\boldsymbol{\rho}_v^\intercal\left(\sum_{j\in\boldsymbol{c}_i}\sum_{v'\in V}\boldsymbol{\alpha}_{v'}\mathbf{x}_{jv'}\right)\right). \tag{1.8}$$

Here, the parameter simply results of the application of the sigmoid function to the dot product between the word embedding $\boldsymbol{\rho}_v$ and the representation of its context for occurrence i.

Dynamic extension. We consider a corpus split into T time slices indexed by t. A word v has T different embedding vectors $\boldsymbol{\rho}_v^{(t)}$, one per time slice, while the context vectors $\boldsymbol{\alpha}_v$ are assumed to be fixed. Figure 1.4 illustrates this system for text data $X^{(1)}, \dots X^{(T)}$. The embedding vector $\boldsymbol{\rho}_v$ drifts throughout time following a Gaussian random walk:

$$\boldsymbol{\rho}_v^{(0)} \sim \mathcal{N}(0, \lambda_0^{-1}I), \quad \text{then } \forall t \geq 1\colon \boldsymbol{\rho}_v^{(t)} \sim \mathcal{N}(\boldsymbol{\rho}_v^{(t-1)}, \lambda^{-1}I). \tag{1.9}$$

The *drift* (precision) hyper-parameter λ controls the temporal evolution of $\boldsymbol{\rho}_v$, and is shared across all time steps.

Training. The training process, described more precisely by Rudolph and Blei (2018), relies on a variant of the negative sampling strategy described by Mikolov, Sutskever, et al. (2013). The goal is to maximise the following objective:

$$\mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\alpha}) = \mathcal{L}_{pos}(\boldsymbol{\rho}, \boldsymbol{\alpha}) + \mathcal{L}_{neg}(\boldsymbol{\rho}, \boldsymbol{\alpha}) + \mathcal{L}_{prior}(\boldsymbol{\rho}, \boldsymbol{\alpha}) \tag{1.10}$$

The first term $\mathcal{L}_{pos}$ is the log-likelihood on positive examples, while $\mathcal{L}_{neg}$ is the log-likelihood computed on a set of randomly selected negative samples. Finally, the role of $\mathcal{L}_{prior}$ is twofold: it acts as a regularisation term on $\boldsymbol{\alpha}$ and $\boldsymbol{\rho}^{(0)}$, and as a constraint on the drift of $\boldsymbol{\rho}$, preventing $\boldsymbol{\rho}_v^{(t)}$ from going too far apart from $\boldsymbol{\rho}_v^{(t-1)}$:

$$\mathcal{L}_{prior}(\boldsymbol{\rho}, \boldsymbol{\alpha}) = -\frac{\lambda_0}{2}\sum_v \|\boldsymbol{\alpha}_v\|^2 - \frac{\lambda_0}{2}\sum_v \|\boldsymbol{\rho}_v^{(0)}\|^2 - \frac{\lambda}{2}\sum_{v,t}\|\boldsymbol{\rho}_v^{(t)} - \boldsymbol{\rho}_v^{(t-1)}\|^2. \tag{1.11}$$

The components of the log-prior depend on the type of regularisation that is required for the model. It is possible to imagine other types of regularisation, for example if one wants the embeddings at t to stay close to the ones at t_0 instead of $t-1$. We introduce several alternatives in chapter 3.

The authors apply their methods to a corpus of U.S. Senate speeches ranging between 1858 and 2009. The word *Iraq* has largest drift in this corpus; a one-dimensional representation of its evolution is shown in Figure 1.5 along with the most similar words for several selected years. The figure shows the smoothness of the trajectory of the embedding, which is enabled by the reg-

ularisation on the prior. In 1858, *Iraq* appears in the same context as other countries and regions such as *Poland*. In the 1950s, the closest words are mostly Arab countries. In 1980, when Iraq invades Iran, most surrounding words belong to the lexical field of war ("troops", "invasion"). Finally, in 2008, words from the lexical field of terrorism appear in the embedding neighborhood of the word.

This was further extended by Jawahar and Seddah (2019), using additional extra-linguistic features — such as spatial, socioeconomic and topic features — to contextualise Bernoulli embeddings and improve the precision of semantic shift detection when dealing with short time periods.

1.5.4 Other methods

We mention here a few other methods that consider the sequential modelling aspect of semantic change detection, by implementing time-sensitive algorithms.

For each time slice t of a corpus, Yao, Sun, Ding, Rao, and Xiong (2018) compute the positive pointwise mutual information matrix ($PPMI^{(t)}$) matrix, of dimension $V \times V$ with V the vocabulary size. Each element of this matrix is computed this way:

$$PPMI(w_1, w_2)^{(t)} = \max\{PMI(w_1, w_2)^{(t)}, 0\} \tag{1.12}$$

The PMI is computed according to equation 1.2. Then, they infer temporal word embeddings $U^{(t)}$ such as $U^{(t)}U^{(t)\intercal} \approx PPMI^{(t)}$. $U^{(t)}$ is learned using a joint optimization problem:

$$\min_{U^{(1)},\dots,U^{(T)}} \frac{1}{2}\sum_{t=1}^{T} \|PPMI^{(t)} - U^{(t)}U^{(t)\intercal}\|^2 + \frac{\lambda}{2}\sum_{t=1}^{T}\|U^{(t)}\|^2 + \frac{\tau}{2}\sum_{t=2}^{T}\|U^{(t-1)} - U^{(t)}\|^2 \tag{1.13}$$

The last term of this equation aims to enforce alignment, pushing each word embedding $u_w^{(t)}$ as close as possible from $u_w^{(t+1)}$. Note that it is very similar to the regularisation of the prior of the DBE model (equation 1.11). The objective function is decomposed across time and solved iteratively using stochastic gradient descent. Like the two previous methods (DBE and DWE), the authors argue that sharing information across time slices is highly beneficial for the quality of diachronic embeddings, especially in situations of data sparsity. This methods is applied to a corpus of New York Times articles published between 1990 and 2016, detecting drifts such as the ones illustrated in Figure 1.6.

Rosenfeld and Erk (2018) propose one of the few models that considers time as a continuous variable. Their model includes three components. First, a derivative of the SGNS algorithm produces a time-independent embedding for each word of the vocabulary. Second, a time embedding components com-

putes a continuous representation of time. Finally, the time embeddings and the time-independent word embedding are combined to create diachronic embeddings. This setup allows to capture the gradual drift of a word with high precision. Moreover, the authors argue that using time representation as a common variable for all word representations allows the model to capture semantic change in a more realistic manner, considering time as a single underlying cause affecting all words similarly.

Finally, Tsakalidis and Liakata (2020) approach semantic change detection as an anomaly identification task, track the evolution of word embeddings through time in a sequential manner. They implement an architecture divided into an autencoder, to learn word representations, and a decoder, to predict the representations of the words at the next time slice. The neural architecture allows to approximate a non-linear function of evolution of word embeddings; when the word embedding predicted using this non-linear function differs from the real one, it means that semantic change has occurred.

1.6 Detection measures and interpretation

The time-varying word representations have to be compared between time slices to measure the level of semantic change of the word. Most of the time, word representations are real-valued vectors; they can be compared using usual distances measures such as Euclidean distance and cosine distance. In particular, the cosine distance (cos) between two embeddings vectors and its opposite, the cosine similarity (1−cos) are widely used metrics in semantic change (Shoemark et al., 2019; Schlechtweg et al., 2019). The cosine distance between the embeddings of two words u_1 and u_2 from a same vector space is given by the following equation, with n the dimension of the embeddings:

$$\cos(u_1, u_2) = \frac{u_1^\intercal \cdot u_2}{||u_1|| \cdot ||u_2||} = \frac{\sum_{i=1}^{n}(u_{1,i} \cdot u_{2,i})}{\sqrt{\sum_{i=1}^{n} u_{1,i}^2} \cdot \sqrt{\sum_{i=1}^{n} u_{2,i}^2}} \tag{1.14}$$

We define two ways of measuring the drift of a word: the *incremental* drift computes the drift from each time slice to the next one, and the *inceptive* drift computes the directed drift from the first time slice $t_0 = 1$.

$$\text{Incremental drift} = \cos(\boldsymbol{u}^{(t)}, \boldsymbol{u}^{(t+1)}) \; \forall t \in [1, T-1] \tag{1.15}$$

$$\text{Inceptive drift} = \cos(\boldsymbol{u}^{(t)}, \boldsymbol{u}^{(1)}) \; \forall t \in [2, T] \tag{1.16}$$

We end up with vectors of drifts of size $T-1$. We can use the sum of their elements to get a unique value quantifying the amount of drift the word has gone through during the whole period. A simpler way to measure it is to only

compute the distance between the embeddings in the first and in the last time slice:

$$\text{Full drift} = \cos(\boldsymbol{u}^{(T)}, \boldsymbol{u}^{(1)}) \tag{1.17}$$

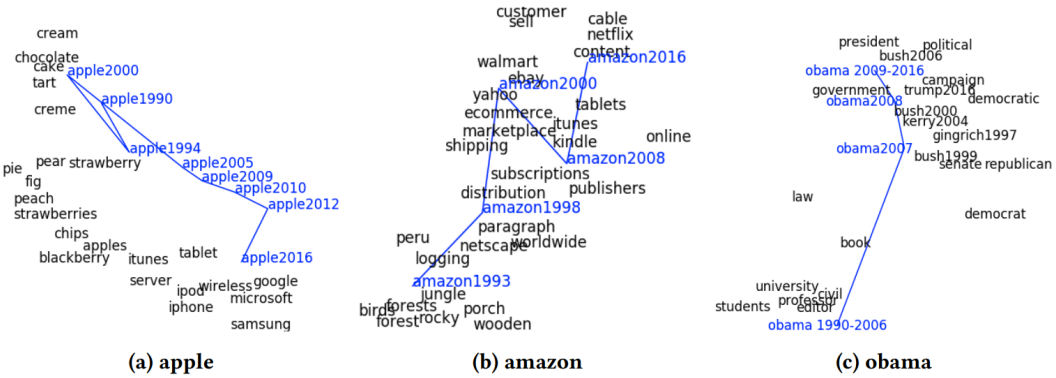
However, using the full time series to compute the drift instead of comparing the first and the last time slice only has been shown to increase the performance of semantic change detection (Shoemark et al., 2019).

On top of quantifying how much a word drifted, a common task is to detect the time slice when a semantic change occurred. To this end, Kulkarni et al. (2015) use a breakpoints detection method. First, they compute the time series of inceptive drifts of the embedding of a word (Equation 1.16). They normalize it, and detect breakpoints using the mean shift model (Taylor, 2000): the normalised series is modeled as the output of a stochastic process. For each time period t, they compute the difference between the means of the series before and after t. It leads to a mean shift series of size $T - 1$. The breakpoints are the significant jumps in this mean shift series. To detect them, they use Monte Carlo permutation tests to estimate the statistical significance of mean-shift scores: the breakpoints occurs when the p-value is minimal.

After detecting the time of the change and quantifying it, it is possible to conduct an interpretation of how words change; in what direction, from which sense to which one. Most methods relying on word embeddings identify the most similar terms of a word at each time slice; it allows to illustrate the general meaning of the word in a period, and deduce how it changed when the most similar terms are different. Figures 1.3 and 1.5 from previous sections and Figure 1.6 illustrate this method. The latter shows the evolution of three target words from 1990 to 2016 in the New York Times newspaper: *apple*, *amazon* and *obama*. A 2-D t-SNE projection allows to highlight the path of the target word in the representation space, and its localisation at a given period can be interpreted using the nearest neighbors in the representation space at that time. We focus on the shift of the word *apple* during the 27-years period, moving from the lexical field of fruits and desserts towards the space of technology. The gradual drift starts in early 2000s when Steve Jobs becomes CEO, but the method identifies a spike in 1994 in the trajectory, related to a notable even for the company *Apple* that had a large press resonance (the replacement of the CEO and a collaboration with IBM).

1.7 Sense-disambiguating methods

Before the rise of pre-trained language models to encode contextual information into word representations, several works made use of the context of a word to disambiguate their sense in a sentence, improve their representation, and detect semantic change. In this section, we give an overview of several



(a) apple **(b) amazon** **(c) obama**

Figure 1.6 – Trajectories of brand names and people through time: apple, amazon and obama (Yao, Sun, Ding, Rao, & Xiong, 2018).

sense-disambiguation methods applied to semantic change detection.

Contextual entropy. Tang, Qu, and Chen (2016) consider the evolution of the contextual entropy of a word across time. The goal is to detect the broadening and narrowing of the set of meanings of a word, and the appearance of a new meaning. They argue that nouns are the most crucial elements for sense identification. Thus, they represent a word with only the strongest associated noun in each context where it appears. The dimension of the co-occurrence matrix is reduced to the list of most closely related nouns to each word in all its contexts. They obtain the distribution over the nouns for each word at each period, and use it as the set of senses associated with the period. The distribution of senses $s_1, s_2, \dots s_n$ of word w at time t is written:

$$p(s_i|w) = \frac{\text{count}_w(s_i)}{\text{count}(w)}. \tag{1.18}$$

With $\text{count}_w(s_i)$ the number of times the noun s_i is strongly associated with w. Then, they compute what they call the "Word Status", which is the entropy of this distribution of nouns:

$$S_w^t = -\sum_{i=1}^{n} p(s_i|w) \log p(s_i|w) \tag{1.19}$$

They observe variations of Word Status over time, which mirror changes in the usage of the word, and deduce the type of change it went through.

Topic models. Lau, Cook, McCarthy, Newman, and Baldwin (2012) and Cook, Lau, McCarthy, and Baldwin (2014) use topic modeling for word sense induction. They associate it with a method to detect novel senses between two periods. In short, the authors apply Hierarchical Dirichlet Process—a variant of Latent Dirichlet Allocation (Blei, Ng, & Jordan, 2003) that automatically selects the optimal number of clusters—to the different time slices indepen-

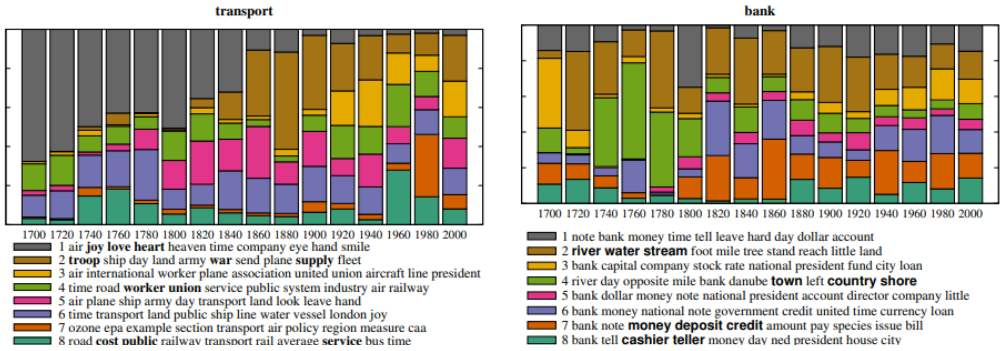


Figure 1.7 – Distributions of topics for the words *transport* and *bank* over 20-year intervals between 1700 and 2010, along with top 10 words for each sense (Frermann & Lapata, 2016).

dently. Then, all topics are pooled and each occurrence of a target word is assigned a topic; each topic being considered as a distinct word sense. They compute a *novelty scores* to determine whether a topic is considered novel, depending of its distribution in the different time slices normalized by a maximum likelihood estimate: a high score means a rare sense in old periods and frequent in recent ones. Note that there is no alignment of senses over time.

Frermann and Lapata (2016) propose the SCAN model, relying on a dynamic topic model (Blei & Lafferty, 2006). The strength of dynamic topic models is that both the distribution of topics in documents and the topics themselves can evolve through time. Each topic is considered as a sense. They are expressed as a distribution over all words of the vocabulary. The distribution of senses in the corpus evolves, and the senses themselves undergo more subtle intrinsic change, visible through the evolution of the words associated with them. The authors define logistic-normal priors for the multinomial sense distributions and the word distribution of each sense. The parameters enforce smooth and gradual changes in topics and words distributions. An additional parameter over word distributions controls the extent of change, forcing the topics to stay thematically consistent and enabling topic comparison across time.

Figure 1.7 shows an example of the evolution of topics distribution over time for two words, *transport* and *bank*. The former sees the emergence of plane-related transportation (topics 2 and 3) and the death of the old sense related to joy and happiness. The distribution of senses for the second word *bank* is stable across time, but its different senses appear clearly in the keywords of the topics ("river bank" versus "monetary institution").

On top of tracking the distribution of senses, the intrinsic evolution of a sense can be identified. One of the examples given by the authors is the word *mouse*. The sense related to the animal stayed stable during the past decades, while the computer-related sense undergone change, from being close to words like *cable*, *ball* and *mousepad* towards words like *optical*, *laser* and *usb*.

Word sense disambiguation methods. Word sense disambiguation provides precise information on the senses of a word. It can be applied to a limited number of target words, but provides valuable information for semantic change detection. The two following methods rely on clustering to extract word senses and track their evolution over time.

Mitra et al. (2015) build a co-occurrence graph by keeping only the closest related words from all bigrams of a corpus. Then, clustering is performed on the graph at each period; each cluster is assumed to correspond to a particular sense of the word. The clusters are aligned over time by building an intersection matrix of words of the graph that appear in the clusters of two time slices. The intersection matrix enables to compare the word overlap of clusters between two periods and deduce the type of meaning drift, classified into categories (birth, death, join or split) according to a set of rules.

Tahmasebi and Risse (2017a) identify word senses at different periods using an unsupervised word sense induction algorithm called curvature clustering (Dorow, 2005). Then, they group all the senses into semantically coherent clusters and track their evolution across time. To group them, they compare the senses using a variant of the Jaccard similarity measure and a WordNet-based similarity measure. In case of high similarity, senses from two consecutive periods are merged into a bigger cluster. This leads to a set of coherent senses (*units*) for each word. Finally, these units are grouped into *paths*, by comparing each unit with all the units of later periods. The authors analyse the evolution of the polysemous concepts through these paths.

1.8 Conclusion

We presented a list of methods to learn diachronic word representations. They are summarised in a timeline in Figure 1.8. Shoemark et al. (2019) an Schlechtweg et al. (2019) compare several of the most common methods presented in this chapter on the task of semantic change detection, using real-world annotated corpora as well as generated corpora with synthetic semantic change. They conclude that methods based on neural word embeddings, in particular the SGNS model, outperform methods relying on PPMI or co-occurrences matrices. A recent evaluation campaign, "Semeval 2020 Task 1: Unsupervised Lexical Semantic Change Detection" [2] shows that the best performing method seems to be, in the case of corpora divided into two disjoint time slices, the SGNS model trained independently on the two time slices and aligned using Orthogonal Procruste, before comparing word vectors using Cosine Similarity (Schlechtweg, McGillivray, Hengchen, Dubossarsky, & Tahmasebi, 2020).

However, diachronic word embeddings methods are not free from defects. Alignment-based methods are source of noise and require a lot of data; incre-

[2] https://competitions.codalab.org/competitions/20948

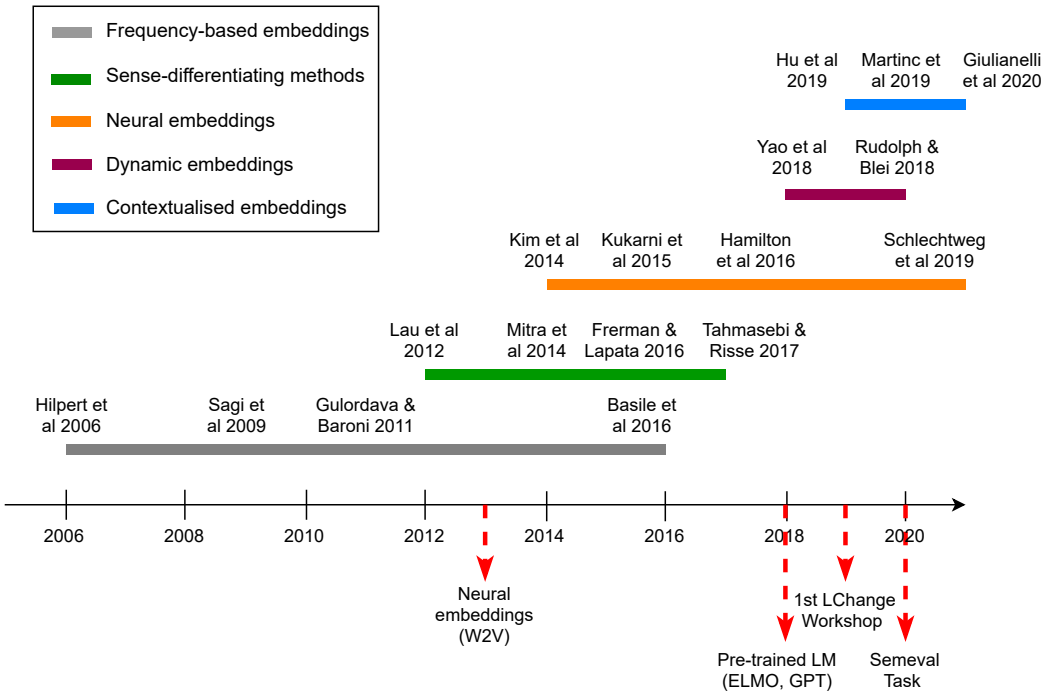


Figure 1.8 – Timeline of the literature of different types of models for diachrony.

mental methods involve a lot of training updates; and dynamic embeddings that use all time slices jointly have the drawback of leaking data from the future towards the past. All of them, overall, are strongly influenced by frequency (Dubossarsky, Weinshall, & Grossman, 2017).

More importantly, most methods presented in this chapter use non-contextual embeddings: each word is represented as a unique vector in a time slice, without taking into account the possibility to have several distinct senses or usages of the word in the corpus. This is clearly illustrated by the first two examples of Figure 1.6. The lexical field of fruits disappears from the surroundings of the word *apple* in the vector space from 2005 and onward. A similar situation happens for the word *amazon*, whose usage as the biggest South America river disappeared after 1998 according to the embeddings.

To solve this issue, one needs a different kind of representation, taking into account the context of the word occurrence to create sense-disambiguated embeddings. Word sense disambiguation methods can be performed to achieve this on a limited set of words (Mitra et al., 2015; Tahmasebi & Risse, 2017a). Recent methods to learn contextualised word embeddings using pre-trained language models such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) gives new perspectives to this problem. However, it involves a completely new way of training, extracting and using word embeddings. In the following chapter, we propose methods for semantic change detection using pre-trained language models.

Chapter 2

Contextualised embeddings for semantic change

Contents

In this chapter, we propose a set of methods that use contextualised embeddings extracted using pre-trained language models to detect semantic change. They are the continuation of the sense-disambiguating methods presented in Section 1.7. Context-aware word embeddings can bring valuable information for the task of semantic change detection. However, they cannot be used the

same way as the non-contextual embeddings presented in the previous chapter. They are extracted from a corpus using a language model, leading to a large number of vectors for each word. Consequently, specific processing steps and measures are necessary to enable comparison of word usage between time slices. In return, contextualised embeddings can help with interpreting semantic change, by identifying which sense appeared or disappeared during a period.

In Section 2.1, we give a quick summary of BERT, the language model we focus on in our experiments. Then, we summarize other works that use pre-trained language models to detect semantic change (Section 2.1.2). Section 2.2 explains how we extract contextualised word embeddings from BERT, before presenting several aggregation methods of contextualised embeddings and semantic change measures. Section 2.3 outlines the limitations of these methods in terms of scalability or interpretability and proposes an improvement of the extraction and aggregation methods to deal with the scalability issue. Finally, section 2.4 sums up the semantic change measures and proposes interpretation methods.

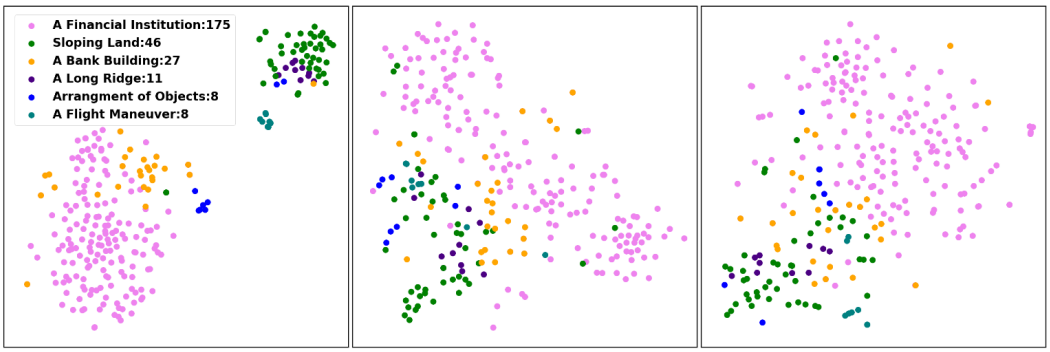
Parts of this chapter were made in collaboration with Matej Martinc, from the Jozen Stefan Institute (Ljubjana, Slovenia) and Elaine Zosa and Lidia Pivovarova, from the University of Helsinki (Finland).

2.1 Introduction to contextualised embeddings

Recent years have seen the emergence of language models pre-trained on large amounts of data. They allow to learn high-quality representation of sentences, leading to improvement of the accuracy for numerous NLP tasks. In particular, their ability to build word representations by exploiting context leads to naturally disambiguated embeddings: the representation of a word contains information about the whole sentence in which it appears.

2.1.1 Pre-trained language models

The new line of work relying on pre-trained language models outperforms classical word embeddings by a significant margin in most NLP tasks. With non-contextual word embedding models, each word from a predefined vocabulary is represented as a unique vector. Contextualised embeddings generate a separate vector for each word mention, i.e. for each context the word appears in. Two widely used contextual embeddings models are ELMo (Embeddings from Language Models, Peters et al., 2018) and BERT (Bidirectional Encoder Representations from Transformers, Devlin et al., 2019).



(a) BERT (b) Flair (c) ELMo

Figure 2.1 – T-SNE plots of contextualized embeddings of *bank* with their WordNet sense, with the frequency of each sense in the SensEval-3 training data, for 3 language models (Wiedemann, Remus, Chawla, & Biemann, 2019).

Presentation of BERT. As an example, we focus on the pre-trained language model BERT (Devlin et al., 2019). Wiedemann, Remus, Chawla, and Biemann (2019) showed its superiority in disambiguating word senses compared to two other models, ELMo (Peters et al., 2018) and Flair NLP (Akbik, Blythe, & Vollgraf, 2018) (see example in Figure 2.1, with the contextualised embeddings of the word *bank* for these 3 models).

BERT is a method for pre-training language representations. Its architecture is a multi-layer bidirectional Transformer encoder (Vaswani et al., 2017), a recent and popular attention model, originally applied to machine translation before being adapted for language modelling.

The key element to this architecture is the bidirectional training that differs from previous approaches, which relied on unidirectional (left to right) or a combination of left-to-right and right-to-left training. It is enabled by a new training strategy, Masked Language Model: 15% of the tokens in each input sequence are selected as training targets, of which 80% are replaced with a [MASK] token. The model is trained to predict the original value of the training targets using the rest of the sequence. This allows BERT to leverage both left and right context, meaning that a word w_t in a sequence is not determined just from its left sequence $w_{1:t-1} = [w_1, \ldots, w_{t-1}]$—as is the case in the traditional causal language modelling task—but also from its right sequence $w_{t+1:n} = [w_{t+1}, \ldots, w_{t+n}]$. A second training strategy is used, named Next Sentence Prediction (NSP): pairs of sentences are generated for input, with 50% beings pairs of successive sentences extracted from a document, and 50% being two random sentences from the corpus. The model is trained to predict if the two sentences are consecutive or not[1].

BERT is mostly used in the literature following the principle of transfer learning proposed by Howard and Ruder (2018), where the network is pre-

[1]In practice, this second training strategy is not used anymore, as it was shown that it does not really improve the quality of the model (Liu et al., 2019).

trained as a language model on large corpora in order to learn general contextualised word representations. This is usually followed by a task-specific fine-tuning step such as text classification. Fine-tuning a language model this way leads to significant improvements of the accuracy on many tasks compared to previous state-of-the-art models (Devlin et al., 2019; Peters, Ruder, & Smith, 2019). In our case, we perform domain adaptation by fine-tuning BERT with the masked language model task on our corpus. We use BERT as a feature extractor: applied to any text sequence, the features extracted for language modeling by BERT can be used as contextualised representation of the input tokens.

2.1.2 Language models for semantic change detection: other works

Using contextualised embeddings for semantic shift detection is fairly novel; we are only aware of few studies that employed them.

In Hu, Li, and Liang (2019), contextualised embeddings are applied in a controlled way: for a set of polysemic target words, a representation for each sense is extracted using a pre-trained BERT model applied to sense-disambiguated sentences. This model is then applied to a diachronic corpus, extracting contextualised embeddings, that are matched to the closest sense embedding. Finally, the proportions for each sense are computed at each successive time slice, revealing the evolution of the distribution of senses for each target word. This method requires that the set of senses of each target word is known beforehand.

Martinc, Novak, and Pollak (2020) propose to average all contextualised embeddings of a word at each time slice in order to generate a time-specific word representation. BERT embeddings are used in their study, with the cosine distance to measure the distance between word representations in different time periods.

Giulianelli, Del Tredici, and Fernández (2020) use k-means clustering on BERT contextualised embeddings and compare several measures to quantify changes between word usages in different time periods. In addition, the incremental training approach proposed by Kim et al. (2014) was used for diachronic fine-tuning of the model; however this yielded no performance improvements.

Finally, Kutuzov and Giulianelli (2020) and Rodina, Trofimova, Kutuzov, and Artemova (2020) conduct similar experiments using the pre-trained language model ELMo (Peters et al., 2018), which was one of the first model to produce contextualised word embedding. It is outperformed by BERT on most NLP tasks, but its lighter architecture—a two-layer Bidirectional LSTM on top of a convolutional layer—and lower number of parameters enables faster training and inference.

2.2 Contextualised embeddings for diachrony: aggregation methods

In this section, we start by explaining how we extract the contextualised embeddings of a word from a corpus. In order to quantify the degree of semantic change of this word, we want to compare its embeddings between the different time slices. Thus, given a time slice, we must summarize the information contained in all the contextual embeddings of a word, in such a manner that it is comparable to other time slices. In this section, we present four methods to achieve this goal.

2.2.1 Extracting contextualised embeddings

We assume that we have a BERT language model pre-trained on a large amount of data, and a corpus divided into time slices. Using the Masked Language Model task, we fine-tune the model on the corpus to get domain-specific knowledge and to increase the quality of the contextualised representations. Note that we do not conduct any diachronic fine-tuning of the model using the incremental training approach described in Section 1.5. The hypothesis is that this step is not necessary due to the contextual nature of embeddings generated by the model, which by definition are dependent on the context that we assume to be time-specific. We only fine-tune the model on the full corpus for domain adaptation.

In order to acquire contextualised embeddings, the corpus documents are first split into sentences. We apply the BERT tokenization which is based on the WordPieces system (Wu et al., 2016), similar to byte-pair encodings. Each sentence is limited to 512 tokens (the usual size limit for a sequence in BERT) and fed into the BERT model. A sequence of embeddings is generated for each of these sequences by summing the last four encoder output layers of the model[2], as semantic features are captured in higher layers of BERT (Devlin et al., 2019; Jawahar, Sagot, & Seddah, 2019). An overview of this process can be found in Figure 2.2. With L being the sequence length and H the dimension of the embeddings (768 in the case of BERT), the output of size $L \times H$ is split along the first dimension to get a separate contextualized embedding for each token in the sequence.

The system of word-pieces raises an additional challenge: we get contextualised representations of subwords as output, whereas we are interested in contextualised representation of full words. To get a word vector from the BERT output of its constituting subwords, various approaches are used. We can concatenate the byte-pairs embeddings together, or take only the first one, which often bears the largest part of the information. We choose to take the av-

[2]We refer the reader to the original description of the transformer in (Vaswani et al., 2017) for a detailed overview of each component in the architecture.

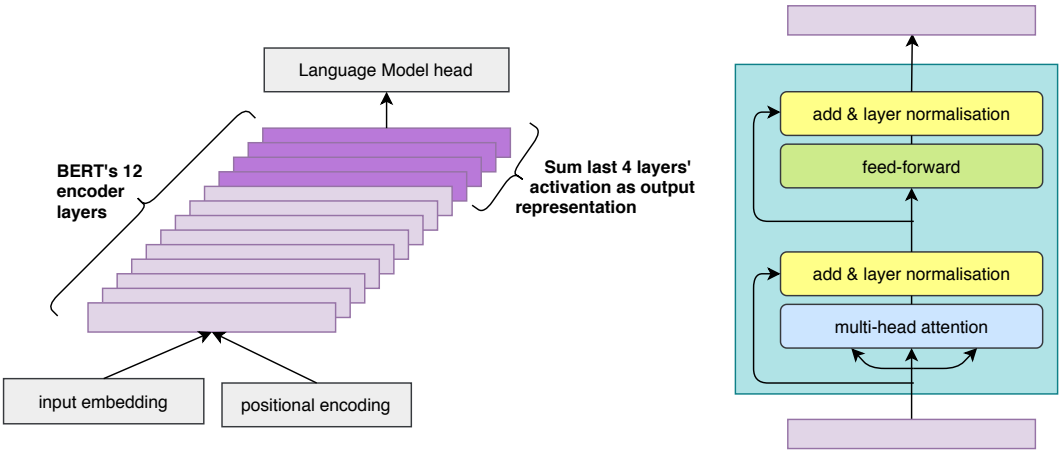


Figure 2.2 – Overview of BERT architecture (on the right) and extraction method for contextualised embeddings (on the left).

erage of all the subwords embeddings as representation for the word (Martinc, Novak, & Pollak, 2020).

2.2.2 Solution 1: no aggregation

A first distance measure can be computed without aggregating the contextualised embeddings of each time slice: the Average Pairwise Distance (APD), also used by Giulianelli et al. (2020). It involves computing the average pairwise distance between all contextualised embeddings of a word in two periods. We note $E_w^{(t)}$ the embeddings matrix of word w at time t, with $N^{(t)}$ its dimension (the number of occurrences of the word in time slice t):

$$APD(E_w^{(t_1)}, E_w^{(t_2)}) = \frac{1}{N^{(t_1)}N^{(t_2)}} \sum_{\substack{u_i \in E_w^{(t_1)} \\ u_j \in E_w^{(t_2)}}} d(u_i, u_j) \tag{2.1}$$

The distance d can be the cosine distance introduced in Section 1.6. We also experiment with the Canberra distance (Lance & Williams, 1967), somewhat similar to the Manhattan distance:

$$d(u, v) = \sum_i \frac{|u_i - v_i|}{|u_i| + |v_i|} \tag{2.2}$$

2.2.3 Solution 2: averaging

Proposed by Martinc, Novak, and Pollak (2020), this method averages all the contextualized embeddings of a word appearing at a given time period. We end up with a set of time-specific vector representations of a word. They can also be compared using the cosine distance, as described in Section 1.6.

$$Avg(E_w^{(t_1)}, E_w^{(t_2)}) = d\left(\frac{\sum_{u_i \in E_w^{(t_1)}} u_i}{N^{(t_1)}}, \frac{\sum_{u_j \in E_w^{(t_2)}} u_j}{N^{(t_2)}}\right) \tag{2.3}$$

2.2.4 Solution 3: clustering

This method groups the set of contextualised embeddings of a word into types of usages. We apply a clustering algorithm to all the contextualised embeddings of a word, on all the time periods jointly, experimenting with *k-means* (Giulianelli et al., 2020) and *affinity propagation* (Martinc, Montariol, et al., 2020a).

In the case of k-means, the user has to define the number of clusters himself. When studying one word in particular, we use the silhouette score (Rousseeuw, 1987) to select the best number of clusters; in practice, it usually ranges between 4 and 7. The silhouette score measures the density of the clusters and the distance between them, by averaging for all points a combination of its mean intra-cluster distance and mean nearest-cluster distance. When doing broader experiment on a large number of target words, we arbitrarily set the number of clusters for all words.

On the contrary, affinity propagation automatically infers the number of clusters during training. This is very convenient for our task, where the number of clusters—the number of different usages of the word—can vary a lot depending on the word, and does not necessarily reflect the number of senses of the word in a lexicographic point of view. Indeed, BERT does not only retain semantic information for contextualised representations; for example, it is heavily influenced by syntax (Reif et al., 2019). The clusters obtained from the representations of a word do not naturally reflect the different senses of the word; they only reflect the different ways it is used.

Affinity propagation (Frey & Dueck, 2007) is an iterative graph-based clustering algorithm. The main idea is that data points exchange messages until a high-quality set of *exemplars*, i.e. members of the input set that are representative of clusters, is obtained. The examplars are chosen according to the criteria of **responsibility** $r(i, k)$, which quantifies how well-suited element k is to be an exemplar for element i, and **availability** $a(i, k)$, which quantifies how appropriate is it for i to choose k as its exemplar. Eventually, clusters of similar points emerge. It has been extensively used in tasks similar to ours, such as word sense induction (Alagić, Šnajder, & Padó, 2018; Kutuzov,

Kuzmenko, & Pivovarova, 2017). The clustering is usually skewed: a limited number of large clusters is accompanied by many clusters consisting of only a couple of instances. It allows for a very precise distinction of the different types of contexts the words appear in; however, it is much harder to provide an interpretation of the meaning of all the different clusters, and of word usage variation in general. On the contrary, k-means tends to produce more even clusters; however, we can still see the appearance of small clusters containing only few instances which do not represent a specific sense or usage of the word.

Merging clusters. We wish to overcome the drawbacks of the clustering method: to decrease the number of clusters a posteriori, in order to focus on the "main" usages of the words while limiting the loss of information. We propose a solution to merge and filter clusters inspired by Amrami and Goldberg (2019).[3] We refer to this method as cluster merging.

First, we consider a cluster to be a *legitimate* representation of the usage of a word if it contains at least 10 instances. The threshold of 10 was derived from the procedure of manual labelling employed in the SemEval 2020 Task on unsupervised lexical semantic change detection (Schlechtweg et al., 2020), where the authors enforce a constraint that each sense must be attested at least 5 times in a time slice in order to be considered as a legitimate sense of a word. In our case, we set the limit that a cluster should contain 10 instances in all time periods, since most of our experiments are done on corpora with two time periods.

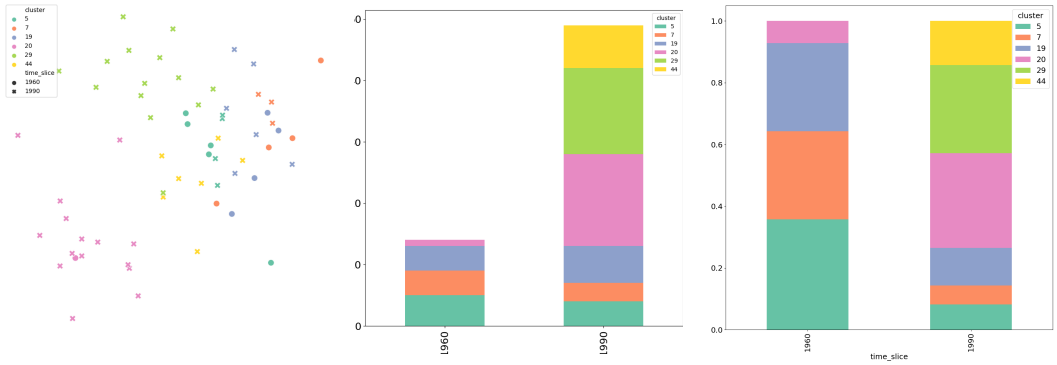
For each cluster, we compute its representation by averaging all embeddings inside the cluster. Then we measure the cosine distance with all other clusters. If the smallest distance is lower than a threshold, the clusters are merged. The threshold is defined as $avg_{cd} - 2 \times std_{cd}$, where avg_{cd} is the average cosine distance between legitimate clusters and std_{cd} is the standard deviation of cosine distances between clusters. For all illegitimate clusters, if there is no cluster that is close enough to be merged with, the cluster is removed. This merging procedure is applied recursively until the minimum distance between the two closest clusters is larger than the threshold. The procedure is also stopped if only two clusters are left.

To sum up, we merge similar clusters together and we delete small clusters if they are not close enough to another cluster to be merged with.

Comparing distributions. After the clustering, all the occurrences of a word are distributed into clusters. We take the example of the word "user" in two time periods, 1960 and 1990; its embeddings are distributed into 6 clusters (Figure 2.3.a). Each embedding is labelled by the time slice where

[3]Note that the procedure of Amrami and Goldberg (2019) is more complex: they first find one or more representatives for each datapoint and then clustering is applied over representatives, while in our work clustering is done over the instances themselves.

it appears. Thus, we can compute the distributions of the clusters C for each period T, that we note $p(C|\mathcal{T}, W)$. More precisely, for a given word, we extract the number of embeddings in each cluster and for each period (Figure 2.3.b); we normalise it by the total number of occurrences of the word in the corpus. We obtain the probability distributions of the usages of this word at each time slice. Figure 2.3.c shows side-by-side the normalised distributions $p(C|\mathcal{T} = 1960, W = user)$ and $p(C|\mathcal{T} = 1990, W = user)$.



(a) Clustering (PCA) (b) Cluster counts (c) Normalised distrib.

Figure 2.3 – Clustering methods for the word *user* in two periods, 1960 and 1990, using affinity propagation.

These distributions can be compared between two periods using the Jensen-Shannon divergence (JSD, Lin, 1991), a measure of similarity between two probability distributions. It is the symmetrization of the well-known Kullback-Leibler (KL) divergence, and can be formally defined as follow:

$$\mathrm{JSD}(d_1, d_2) = \frac{D_{KL}(d_1 \| M) + D_{KL}(d_2 \| M)}{2}, \tag{2.4}$$

with $M = \frac{d_1 + d_2}{2}$.

For example, to quantify how much the usage of the word w changed between the time slices t_1 and t_2, we can compute $\mathrm{JSD}(p(C|t_1, w), p(C|t_2, w))$.

The generalisation of the JSD to n probability distributions $d_1, d_2, \ldots, d_n$ (Ré & Azad, 2014) can be used to compare more than 2 periods together. With H being the Shannon entropy function, the generalised JSD is defined as:

$$\mathrm{JSD}(d_1, d_2, \ldots, d_n) = \mathrm{H}\left(\frac{\sum_{i=1}^{n} d_i}{n}\right) - \frac{\sum_{i=1}^{n} \mathrm{H}(d_i)}{n} \tag{2.5}$$

The JSD is high when the compared distributions assign different probabilities to the same clusters.

2.2.5 Solution 4: optimal transport

The averaging method retains the original dimension of BERT embeddings, 768. It allows very precise comparison of the average context of a word between two time slices. But it loses all information about the diversity of the embeddings inside the time slices. On the contrary, the clustering method sums up the variability of the context of a word in a time slice by breaking it down into a distribution of very small dimension. However, it loses all of the fine-grained semantic information learnt by the model and saved into the embeddings. To keep both types of information to compare word usage between two time slices, we rely on the optimal transport framework.

2.2.5.1 Formulation

In order to catch the level of variability of the set of embeddings in a time slice, we start by clustering it. We can either do a unique clustering, like in the previous section, or perform as many clusterings as time slices independently. Then, to reduce the amount of data without loosing too much information, we average all the embeddings inside each cluster. Thus, in a situation with K clusters and T time slices, we end up with a matrix of size $T \times K \times 768$ for each word. We have one additional information from the clustering: the number of embeddings in each cluster is normalised into a distribution, as in the previous section. This distribution can be specified by a $T \times K$ stochastic matrix. Intuitively, we summed up all the information of the embedding cloud in each time slice into K representatives weighted by the number of embeddings inside the associated cluster, and we want to compare these representatives between time slices.

This setup can be formulated the following way. We call $\mu^{(1)}, \mu^{(2)} \in \mathbb{R}^{K \times 768}$ the sets of K representatives in the two periods, and $c^{(1)}, c^{(2)} \in \Delta^{K-1}$ the marginal distributions of the clusters. Using the notations from the previous section, we have $c_i^{(t)} = p(C = i | \mathcal{T} = t, w)$. We denote by Δ^{K-1} the standard $K-1$ simplex: $c^{(1)}$ and $c^{(2)}$ are positive vectors of dimension K and sum to 1. They represent the weights of each representative in the source and target spaces ($\mu^{(1)}$ and $\mu^{(2)}$).

We quantify the effort of moving one unit of mass from a representative in $\mu^{(1)}$ to a representative in $\mu^{(2)}$ using a chosen cost function, here the cosine distance. Then, we solve the problem by looking for the minimal effort required to reconfigure $c^{(1)}$'s mass distribution over $\mu^{(1)}$ into that of $c^{(2)}$ over $\mu^{(2)}$.

2.2.5.2 The Wasserstein distance

An optimisation problem dedicated to solve this is Optimal Transport, also called the Monge–Kantorovich problem. It can be formulated and solved with Linear Programming. Here, we give a short overview of the Optimal Transport

framework and the Wasserstein distance; for more details, we refer the reader to papers such as (Solomon, 2018).

The common example given when introducing Optimal Transport is the pile of sand problem: to quantify the distance between two piles of sand, we calculate the amount of work needed to move one pile into the other. This metaphor gave birth to another name for the Wasserstein distance: *Earth mover's distance* (EMD). The Wasserstein distance W is nonnegative, symmetric, and satisfy the triangle inequality: all properties that make it a proper distance. On our problem, it can be computed the following way (with cos the cosine distance):

$$\begin{aligned} W(c^{(1)}, c^{(2)}) = & \min_{\gamma} \sum_{i,j} \gamma_{ij} \cos(\mu_i^{(1)}, \mu_j^{(2)}) \\ & \text{with } \gamma \mathbf{1} = c^{(1)} \\ & \gamma^\intercal \mathbf{1} = c^{(2)} \\ & \gamma \geq 0 \end{aligned} \tag{2.6}$$

In other words, we want to minimize the total work ($\min_\gamma$) given that the transported mass is non-negative ($\gamma \geq 0$), and that we start from $c^{(1)}$ to go to $c^{(2)}$. Solving this equation leads to a transport plan γ. It can be seen as a probability mass function over $K \times K$ whose marginals are $c^{(1)}$ and $c^{(2)}$, and quantifies what proportion of the mass $c_i^{(1)}$ at $\mu_i^{(1)}$ should be transferred to $\mu_j^{(2)}$ in order to place a mass of $c_j^{(2)}$ there, in the most efficient way. This matrix can be summed to obtain the Wasserstein distance: the sum of all the work needed to solve the problem.

We note that is problem is completely different from the setup from the previous section solved with the Jensen-Shannon divergence; instead of comparing two distributions, we compare two weighted sets of points. That is why we do not need aligned clusters from a unique clustering; two independent clusterings, one per time slice, could allow a better fit for each set of points without harming the computation of the distance.

2.2.5.3 Regularisation: the Sinkhorn algorithm

The Sinkhorn algorithm is an entropic-regularized method; it is the most common regularization strategy for optimal transport (Cuturi, 2013). We use the same notation as for Wasserstein distance, and add a regularization term

weighted by $\lambda > 0$:

$$\begin{aligned} \text{Sink}(c^{(1)}, c^{(2)}) = \min_{\gamma} \sum_{i,j} \gamma_{ij} \cos(\mu_i^{(1)}, \mu_j^{(2)}) - \frac{1}{\lambda} \text{H}(\gamma) \\ \text{with } \gamma \mathbf{1} = c^{(1)} \\ \gamma^\intercal \mathbf{1} = c^{(2)} \\ \gamma \geq 0 \end{aligned} \tag{2.7}$$

Here, H is the entropy function written the following way:

$$\text{H}(\gamma) = \sum_{i,j} \gamma_{ij} \log(\gamma_{ij}) \tag{2.8}$$

Controlling the regularisation amounts to tuning the parameter $\lambda > 0$. As λ grows bigger, the Sinkhorn formulation converges to the Wasserstein distance. A large λ leads to less numerical stability, and a small value of λ can out-perform the Wasserstein distance. In our experiments, we start with $\lambda = 10^{-3}$ and increases it ten-folds until $\lambda = 10$.

2.3 Scalable extraction and aggregation

The different solutions for quantifying semantic change presented in the previous section all have their limitations. They involve to save in memory all the contextualised embeddings of each time slices (solutions 1, 3 and 4), to cluster a large number of embeddings (solutions 3 and 4), or to forego the differenciations of the different senses provided by the clustering, thus loosing important information for interpretation (solutions 1 and 2). In this section, we further explain these different limitations and propose two methods to overcome them.

2.3.1 Scalability and interpretability limitations

The main limitation of the clustering and Average Pairwise Distance methods is the scalability in terms of memory consumption and computational time, since the clustering needs to be applied to each word in the corpus separately and since all occurrences of a word need to be saved in memory. For large corpora with large vocabularies, where some words can appear millions of times, the usability of these methods is severely limited.

2.3.1.1 Complexity and memory consumption of clustering

The scalability issues of the methods can be divided into two separate problems.

Memory consumption. The token embeddings of each target word are aggregated into lists, on which clustering or Average Pairwise Distance is applied. This procedure becomes unfeasible for a large set of target words or if the embeddings are generated from a large corpus, since too many embeddings need to be saved into memory for further processing. To give an example, a single-precision floating-point requires 4 bytes of memory and each contextualised embedding contains 768 floats (Devlin et al., 2019), meaning that each embedding occupies 3072 bytes if we ignore the additional memory of a Python container (e.g., a Numpy list or a Pytorch tensor) required for storing this data. If we want to use the clustering method on corpora of large size such as the Aylien Coronavirus News Dataset[4] which contains about 250 million tokens, about 768 GB RAM would be necessary to store the embeddings for the entire corpus. If we limit our vocabulary to the 10 000 most common words, we need to generate contextualised embeddings for 230 million tokens, which is still about 707 GB of RAM.

Complexity of clustering algorithms. For the complexity analyses, we denote by d the dimension of the embedding, k the number of clusters if predefined, and n the number of contextualised embeddings i.e. the number of word usages in the corpus. The time complexity of the affinity propagation algorithm is $O(n^2td)$, with t being the predefined maximum number of iterations of the data point message exchange. The time complexity of the simpler k-means algorithm[5] can be stated as $O(tknd)$, where t is the number of iterations of Lloyd's algorithm (Lloyd, 1982) for finding uniformly spaced sets of points in subsets of Euclidean spaces and partitions of these subsets into evenly sized convex cells.

To give an example of what this means for clustering of contextualised embeddings of frequent words, we can look at a widely used corpus for semantic change detection: Google Books. It includes about 4% of all books ever published. The word *rocker* appears in this corpus about 1 million times. For k-means with $k = 5$ and a maximal number of iterations set to 300 (which is the default in `sklearn`), about $300 \times 5 \times 1\,000\,000 \times 768 = 1.152 \times 10^{12}$ operations need to be conducted for the clustering. With affinity propagation with the maximum number of iterations set to 200 (the default value), clustering of the word *rocker* would require $1\,000\,000^2 \times 200 \times 768 = 1.536 \times 10^{17}$ operations, which is impossible to conduct in a reasonable amount of time even

[4] https://blog.aylien.com/free-coronavirus-news-dataset/

[5] Here we are referring to the Scikit implementation of the algorithm employed in this work: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

on a high-performance cluster. The computational power needed to study the evolution of the word *people*, which appears almost 660 million times in this corpus, would be even more unreasonable.

2.3.1.2 Motivation for scalability

To avoid the scalability issue, the methods based on clustering of contextualised embeddings in the literature are only applied to a small set of manually pre-selected words, usually around one hundred (Giulianelli et al., 2020; Martinc, Montariol, et al., 2020a). In practice, this limitation drastically reduces the possible applications of the methods, such a:

- **Detection of the most changed words:** Identification of words that undergo the biggest semantic change is limited to a small manually predefined subset of words for which the change has been somewhat expected in advance. It prevents from detecting unexpected semantic change and notice associated social and cultural phenomena.
- **Group comparison:** Comparison of the rate of semantic change of different categories of words (e.g., semantic categories corresponding to news topics, different word types...) becomes problematic due to small sample size, since a larger set of words would be required for successful generalization.
- **Measuring semantic change of frequent words:** Words that appear very often in the corpus—e.g., more than hundred thousand times—are hard to cluster due to quadratic time complexity of some clustering algorithms, such as affinity propagation (Frey & Dueck, 2007), and are therefore discarded or need to be randomly down-sampled.

Alternatively, when using the averaging aggregation method, embeddings for each word are not collected in a list but rather summed together in an element-wise fashion before being divided by the number of occurrences of the word. Only 768 floats and an integer counter need to be saved for each word of a vocabulary, leading to only 30.72 MB of RAM being required to store the embeddings for the 10 000 most common words in a vocabulary. The approach also does not require any clustering step. However, these representations lose a lot of their interpretability: all word usages are merged into a single averaged representation, leading to the loss of a lot of information on the different usages of the words. This makes the method inappropriate for some tasks such as automatic labelling of word senses.

2.3.2 Target words selection

To begin with, we propose a simple solution to cope with the scalability issue: a preliminary step that would be applied on the full vocabulary of a

corpus, to identify the words that may have undergone semantic change. We investigate several scalable metrics.

A first set of metrics relies on the **averaging** method presented in Section 2.2.3. All contextualised embeddings are averaged at each time slice. The set of contextualised embeddings for word w at time t is noted $\mathrm{E}_w^{(t)}$ and their average is noted $\mu_w^{(t)}$. The cosine distance between average embeddings is used as a measure of semantic drift between time slices. The *total drift* is the cosine distance between the average of token representations of the first time slice (at $t_0 = 1$) and of the last time slice:

$$d_w = \cos(\mu_w^{(1)}, \mu_w^{(T)})$$

It represents the amount of change a word has undergone from the first to the last period, without taking into account the variations in between. The *average incremental drift* computes the mean of the drifts from each time step to the next one, in order to measure the successive changes of word usage:

$$D_w = \frac{1}{T} \sum_{t=1}^{T-1} \cos(\mu_w^{(t)}, \mu_w^{(t+1)})$$

The third metric relies on the computation of a *variation* measure, similarly to Kutuzov (2020). The variation is the cosine distance between each contextualised embedding and a centroid, i.e. the average token embedding for a given word. The mean of these cosine distances is the *variation coefficient* of a word. The intuition is that for words that have several different senses and usages, the distance to the centroid would be higher than for words that are monosemous. However, this method does not make the distinction between words that gain or lose senses and polysemous words that stay stable across time.

To measure the evolution of a word's variation, we compute its variation coefficient inside each time slice t, for the set of contextualised embeddings $E_w^{(t)}$ of word w at time t:

$$v_w^{(t)} = \frac{1}{|E_w^{(t)}|} \sum_{u \in E_w^{(t)}} \cos(\mu_w^{(t)}, u)$$

Then we take the average difference between variations at consecutive time slices. This measure aims at detecting words that undergo changes in their level of polysemy. For example, in a corpus divided into T time slices, the global variation coefficient is:

$$V_w = \frac{1}{T} \sum_{t=1}^{T-1} |v_w^{(t)} - v_w^{(t+1)}|$$

Finally, we can choose a threshold (as a fraction of the size of the full vocabulary) to get a reduced list of target words, from the ranked list of all words according to one of these measures. To compute these metrics, we only need to loop over the dataset once or twice while storing one vector per word in the vocabulary. Thus, they can be used to filter the vocabulary before applying heavier and more accurate methods.

2.3.3 Scalable extraction of contextualised embeddings

In this section, we investigate a solution to the scalability problem: a method to cluster contextualized word embeddings, which is a mixture of the clustering-based and averaging-based approaches described in section 2.2. The proposed method offers improved scalability while still generating interpretable representations.

We follow the embeddings extraction method described in the previous section: we feed all sentences containing a given target word into BERT and extract the contextualised embeddings. The difference lies in the selection of contextualised embeddings. At each occurrence, we decide whether the embedding vector should be *saved* in the list of embeddings for the word or *merged* with one of the previously obtained vectors stored in the list. To improve the scalability of the method, we limit the number of contextualised embeddings that should be kept in memory for each target word in each time slice to a predefined threshold. The threshold of 200 was chosen empirically from a set of threshold candidates (20, 50, 100, 200, 500) and offers a reasonable compromise between scalability and performance. The new vector is merged if it is similar enough to one of the saved vectors or if the list already contains the predefined maximum number of vectors (200 in our case).

More formally, we add the new embedding e_{new} to the list of word embeddings $L = \{e_1, \ldots, e_n\}$ if:

$$|L| < 200 \quad \wedge \quad \forall e_i \in L\colon \cos(e_{\text{new}}, e_i) > \varepsilon$$

ε is the threshold for the cosine distance. In practice, we set it to 0.01.

If $|L| \geq 200$ or if any vector in the list L is too similar to e_{new}, we find the element e_m in the list which is the most similar to e_{new} (i.e., which minimize the cosine distance):

$$e_m = \arg\max_{e_i \in L} \cos(e_i, e_{\text{new}})$$

This element e_m is then modified by summing it with e_{new}: $e_m \leftarrow e_m + e_{\text{new}}$. Besides storing 200 word representations for each word in each time slice, for each group of summed-up representations we also store the number of summed-up elements. When adding an embedding to element e_m, simultaneously, the counter associated with e_m is incremented: $c_m \leftarrow c_m + 1$. Once the model has been fed with all the sequences in the time slice, we divide the final summed-up

vector by c_m. We obtain an averaged embedding for each of the 200 elements in the list. Note that the cosine distance to a sum of vectors is the same as to their averaging, thus there is no need to average after each step.

In short, we first collect 200 word vectors avoiding only near-duplicates representations. Once we have obtained a maximum number of vectors, the subsequent vectors are merged to the most similar ones among the 200 representations already collected.

Note that by having only 200 merged word usage embeddings per word per time slice, and by limiting the vocabulary of the corpus to 10 000 target words, we require around 6.14 GB of space for each time slice, no matter the size of the corpus. While this still uses 200 times more space than the averaging method, the proposed method nevertheless keeps the bulk of the interpretability compared with the standard clustering method.

2.3.4 Quantifying temporal shift

After collecting 200 vectors for each word in each time slice, we can perform the same semantic shift detection methods as for the non-scalable embeddings. First, we can compute the Average Pairwise Distance between the sets of 200 embeddings of a word in adjacent time slices. Second, we can perform clustering of the contextualised word embeddings of all time slices jointly, to derive a unique cluster distribution for all occurrences of a word at different time slices. The distributions can be compared using the Jensen-Shannon Divergence.

We can also use the Optimal Transport method, by computing the Wasserstein or Sinkhorn distance between the sets of representatives of each cluster weighted by the cluster distribution. This can be performed with either a unique clustering or one clustering per time slice. Moreover, the 200 embeddings per time slice extracted by the scalable method have an additional information that can be valuable: the count of word occurrences c_m that were merged with each of the 200 embeddings e_m. This information can be used as weights for the averaging of embeddings in each cluster and time slice, to obtain the set of representatives. We call this method *weighted* OT.

Finally, we can use the Optimal Transport method on all pairs of 200 embeddings without performing clustering. The distributions used to weight the cost matrix in the Wasserstein distance are the counts associated with all embeddings. We call this method *full* OT, as it does not require any dimension reduction through clustering and averaging.

2.4 Semantic shift detection and interpretation

We use the information on word usage extracted from the different time slices to deduce two pieces of information:

1. How much does the usage of a given word vary in the corpus?
2. What is the change about, which usages of the word are involved? How to make an interpretation of this change?

Quantifying semantic change. We use the strategies defined in Section 1.6 to measure the drift of a word: the *incremental* drift, from each time slice to the next one, the *inceptive* drift, from the beginning of the period to each time slice, and the *full* drift, between $t_0 = 1$ and the last time slice only.

The distance measure used depends on the type of information extracted from the contextualised embeddings for a given target word. It can be averaged embeddings $\boldsymbol{u}^{(t)}$ at each time slice, distributions of clusters $\boldsymbol{c}^{(t)}$, pairs of average cluster embedding and cluster distribution $(\boldsymbol{\mu}^{(t)}, \boldsymbol{c}^{(t)})$ or raw matrices of contextualised embeddings $E^{(t)}$.

We sum up the semantic shift detection measures introduced in this chapter. Without preforming any aggregation method, the matrices of contextualised embeddings can be compared using the average pairwise distance (APD). For the averaging aggregation method, the averaged embeddings at each time slice can be compared using the cosine distance (cos). For the clustering aggregation method, the divergence between cluster distributions can be measured with the JSD and its generalisation to n probability distributions. For the Optimal Transport method which involve consecutively clustering and averaging embeddings, the difference is computed using either the Wasserstein Distance or the Sinkhorn Divergence. All these measures can be computed with either the standard or the scalable embeddings.

To put it in a more formal way, the drift between two time slices for a given target word is to be evaluated in the following way:

$$d(t_1, t_2) = \begin{cases} \mathrm{APD}(E^{(t_1)}, E^{(t_2)}) & \text{when using no aggregation} \\ \cos(\boldsymbol{u}^{(t_1)}, \boldsymbol{u}^{(t_2)}) & \text{when using averaging} \\ \mathrm{JSD}(\boldsymbol{c}^{(t_1)}, \boldsymbol{c}^{(t_2)}) & \text{when using clustering} \\ \mathrm{WD}(\boldsymbol{\mu}^{(t_1)}, \boldsymbol{\mu}^{(t_2)}, \boldsymbol{c}^{(t_1)}, \boldsymbol{c}^{(t_2)}) & \text{when using optimal transport.} \end{cases} \tag{2.9}$$

The evaluation can be made using different methods depending on the available annotations associated with the corpus. The words can be *ranked*

according to how much they vary along two or more time slices; if a ground-truth ranking exists, the ranking obtained can be evaluated using Spearman's rank correlation to compare with the true ranking. In an exploratory scenario, the ranking is used to detect the most changing words and then investigate the most unevenly distributed clusters over time for interpretation of the change. Sometimes more detailed information is available; for example, if we use a generated corpus with synthetic semantic drifts, the information used for the generation of the drifts can be used as ground truth for evaluation. It might be the distribution of senses of a target word at each time slice; in that case, the vectors of drifts at all time slices can be computed and compared with the incremental drift or the inceptive drift.

Interpretation. Once the most changing words are detected, the next step is to understand *how* they change between two time slices by interpreting their clusters. We want to identify a common usage to all the contextualised embeddings in each cluster.

In order to capture the clusters involved in the variation, we identify the ones that have an uneven distribution across the considered time frame, by looking at $p(C, \mathcal{T}|w)$. It allows for example to find the clusters that vary the most, and the ones that appear or disappear through time.

However, a cluster may contain several hundred or thousands of word usages contextualised in sentences. Interpreting the underlying sense behind each cluster by manually looking at the sentences inside it would be time-consuming. To reduce human work, we use two methods.

First, we identify the centroids of the clusters: the example (in our case, the sentence) that is the closest to the centroid is assumed to be representative of the context of the word occurrences inside the cluster. Thus, we observe these central sentences to get a preliminary idea of the word usages in context.

Second, we set up a keyword detection method to characterise the different clusters in relation to one another. We rely on the tf-idf (Term Frequency–Inverse Document Frequency) method. Each cluster containing a set of sentences, we consider them as documents and the set of clusters as a corpus. The goal is to identify the most discriminant words for each cluster. The stopwords and the words appearing in more than 50% of the clusters are excluded from the analysis, to ensure that we select as keywords only the words that are specific to one cluster. We compute the tf-idf score of each word in each cluster. We obtain a ranked list of keywords for each cluster. The words with the highest score in a cluster are the most important for the analysis of this cluster: they are used as keywords to ease its interpretation.

Selection of the clustering algorithm. The keywords extracted for each cluster will only provide meaningful interpretation if the quality of the clustering is high; otherwise, it would just lead to a spurious interpretation. However,

it is hard to find the best clustering method to apply to the vocabulary since their performance vary across datasets (see Chapter 3). It is even harder to know which clustering method and what number of clusters produce the most meaningful clustering for each specific word.

A preliminary evaluation of the clustering can be made using the silhouette score (see Section 2.2.4). However, it only evaluates the adequacy of a token to its cluster compared to the other clusters. As a complementary evaluation, we propose to use two measures relying on the list of keywords characterising each cluster. These measures come from the topic modelling literature, where each topic is characterised by a distribution over the vocabulary.

First, the Normalized Point-wise Mutual Information (NPMI, Aletras and Stevenson, 2013) measures the coherence between the top k words characterizing each cluster, using word co-occurrences in the set of sentences. A high NPMI measure indicates a coherent clustering, thus easier to interpret.

Secondly, we adopt a topic uniqueness measure (TU, Nan, Ding, Nallapati, and Xiang, 2019). TU is computed using the top k keywords for a given cluster:

$$TU = \frac{1}{k} \sum_{i=1}^{k} \frac{1}{\text{count}(w_i)},$$

where $\text{count}(w_i)$ is the number of times the i-th word in the top list of this cluster appears in the top list of any clusters. We compute the global TU measure for the whole clustering as the mean of the TU of all clusters. The higher the TU measure is (i.e. closer to 1), the higher the diversity of clusters.

We take $k = 10$ top keywords to compute the NPMI and $k = 25$ for TU. We use these measures to select the best clustering method and the optimal number of clusters.

2.5 Conclusion

In this chapter, we proposed two methods to extract contextualised embeddings from a pre-trained language model. We presented these methods for the BERT model, but they can be applied to any pre-trained language model that can produce contextualised word representations. The first extraction method, the *standard* method, extracts and stores the embeddings of all occurrences of a word. We showed the limitations of this method in terms of scalability, for example for situations where we want to identify semantic change in a large vocabulary. Thus, we proposed a *scalable* method to extract only 200 embeddings per word per time slice. It extracts contextualised embeddings in a streaming fashion, and groups them simultaneously.

We also listed four types of methods to quantify semantic change of a word across two periods, using the *standard* or *scalable* extracted embeddings: Av-

erage Pairwise Distance, Averaging + Cosine Distance, Clustering + Jensen-Shannon Divergence, and Optimal Transport with Wasserstein or Sinkhorn distances. These different methods all have their advantages and drawbacks, in terms of computing time, memory consumption and interpretability. Moreover, they all capture different information when processing the contextualised embeddings: the contextual diversity, in the case of APD; the global drift averaged over all possible contexts, in the case of averaging; the increase or decrease of the relative proportion a word sense, in the case of clustering with JSD. In the following chapter, we apply all methods to several corpora using the language model BERT, to compare their behaviour and evaluate their ability to detect semantic change. We also propose a use case of semantic change detection on a corpus of newspaper articles, to showcase the ability of these methods to detect meaningful and interpretable semantic change.

Chapter 3

Evaluation and comparison

Contents

As noted by most authors in this field, evaluating lexical semantic change detection methods on a large scale is notoriously challenging (Frermann & Lapata, 2016; Tahmasebi et al., 2018). It is a difficult task to annotate, resulting in few labeled corpora for evaluation. Many authors resolve themselves to qualitative evaluation only, often through manual interpretation of the detected semantic drifts. Another solution is to generate a corpus with synthetic semantic drifts. At the time of writing, the recent creation of new manually annotated corpora allows researchers to conduct more thorough evaluation.

In this chapter, we apply the different models introduced in Chapter 2 to several corpora in order to compare them, to observe their behaviour and to evaluate them on annotated data. Parts of the experiments in this chapter were made in collaboration with Matej Martinc from the Jozen Stefan Institute (Ljubjana, Slovenia), and Elaine Zosa and Lidia Pivovarova from the University of Helsinki (Finland).

The first section of this chapter is dedicated to a review of the corpora and the evaluation methods used in the literature. We divide evaluation methods into three categories: without manual annotation (including qualitative evaluation), with manual annotations, and with synthetic data.

Then, in Section 3.2, we apply the methods introduced in Chapter 2 and a selection of methods from Chapter 1 to a large corpus of news articles, the New York Times Annotated Corpus. We analyse the drifts identified by the various models and compare their behaviour, deducing their respective pros and cons.

Sections 3.3 and 3.4 introduce 6 annotated datasets for evaluation of semantic change detection. We provide a detailed comparison of all the methods exposed in Chapter 2; we conclude with a comparison with other state of the art approaches and an error analysis.

Section 3.5 exposes some methods and results on a complementary task: instead of measuring semantic change, we attempt to detect whether a word's meaning changes or not in a binary fashion.

Finally, Section 3.6 leaves performance aside to focus on the practical application of semantic change detection, in particular through interpretation. We apply semantic change detection on a large corpus of COVID-related news articles and comment on the results.

3.1 Literature: data and evaluation

A crucial question when building annotations for any task is whether the annotations should be done with respect to the corpus, or to the outside world (Tahmasebi et al., 2018). In the first case, semantic change annotations need

to be performed relatively to what is in the corpus. However, the scope of the analysis is then limited to the extend to which the corpus is representative of language in the outside world. In the second case, external resources such as dictionaries can be used to get information on word senses across time. These resources must be matched with a corpus to make the evaluation. However, such resources can hardly give precise information on semantic change.

3.1.1 Corpora for semantic change detection

The datasets and the granularity we use depend on the kind of semantic change that we wish to exhibit. We can either look for short-term changes—when the corpus spans several months, for example in the case of news articles or tweets—or long-term changes—when the corpus spans decades or centuries, for example in the case of books and literature. The type of text, depending on the source, is also an important factor: one will not look for the same kind of lexical semantic drift in a corpus of business news and in a corpus of Reddit comments, even though they have the same granularity. Here, we give a few examples of the most common corpora for different granularities.

A low-granularity corpus is the Helsinki corpus.[1] It includes documents of multiple genres from Old, Middle and Early Modern English (from 850 to 1710), for a total of around 1.5 million words, to study the history of English under the scope of morphology and syntax on top of semantics.

A very widely used corpus is Google Books Ngrams [2], released in December 2010. It consists of sub-corpora in several languages; on top of English, there are for example Chinese, French, German, Hebrew corpora... Each corpus is organized as n-gram counts per year, with $n = 1$ to 5, and is constructed using all the books digitized in Google Books. The English n-grams corpus cover about 5% of all books ever published. It was used in the literature of computational linguistics to study semantic change, dividing it into yearly time spans (Bamler & Mandt, 2017) or decades (Gulordava & Baroni, 2011; Mitra et al., 2014). However, it is criticised for its limitation to 5-grams, preventing the analysts to extract large contexts to study word usage. Moreover, the presence of OCR errors decrease the quality of the corpus, and it is not genre-balanced across time. For example, the proportion of scientific literature increases across time and may lead to biased observations (Pechenick, Danforth, & Dodds, 2015).

A corpus designed for historical linguistics is the Corpus of Historical American English (COHA, Davies, 2012). It contains more than 400 million words from 1810 to 2000. As a historical corpus, it is smaller than the Google books corpus but it has the advantage that data from each decade are balanced by genre—fiction, magazines, newspapers, and non-fiction texts, gathered from various Web sources. Its contemporary counterpart, the Corpus of Contempo-

[1] http://korpus.uib.no/icame/manuals/HC/INDEX.HTM
[2] http://storage.googleapis.com/books/ngrams/books/datasetsv2.html

rary American English (COCA, Davies, 2010) spans from 1990 to 2017. It is also genre-balanced and contains more than 560 million words.

Also used for short-term semantic change analysis, the *New York Times Annotated Corpus* (NYT[3], Sandhaus, 2008) collects articles from the New York Times from 1987 to 2006. It has been used a lot in the literature of diachronic word embeddings. Szymanski (2017) and Zhang et al. (2015) rely on this corpus to solve a task of temporal analogies, while Azarbonyad et al. (2017) use it to analyse the references to terrorism and Islam before and after the World Trade Center attacks. Yao et al. (2018) base their analysis on a New York Times corpus similar to this one, but extracted from the newspaper's API over a longer period (27 years).

For very short-term diachrony, corpus of tweets have been used in the literature (Kulkarni et al., 2015; Bamler & Mandt, 2017; Shoemark et al., 2019; Jawahar & Seddah, 2019).

Finally, more domain-specific corpora have been studied: Amazon Movie Reviews (Kulkarni et al., 2015), political corpora such as the State of the Union addresses[4] (Bamler & Mandt, 2017) or U.S. Senate speech (Rudolph & Blei, 2018), or scientific corpora such as ACM abstracts and Machine Learning papers from ArXiv (Rudolph & Blei, 2018).

In this section, we mostly described English corpora. More examples of corpora, as well as corpora in other languages, can be found in Tahmasebi et al. (2018).

3.1.2 Evaluation methods

The study of semantic change faces the issue of evaluation. Very few labeled corpora exist, as it is a task notoriously difficult to annotate. Evaluation can be done using several types of resources: (i) indirect information, usually external resources such as dictionaries; (ii) direct (manual) annotation of semantic change or semantic relatedness in a corpus; or (iii) by generating a corpus with synthetic semantic drifts.

Regardless of the presence or absence of any type of annotation for the task of semantic change detection, all corpora are annotated with the period when each document was written.

3.1.2.1 Corpora without manual annotations

To evaluate semantic change detection methods without semantic change annotations, one of the first things we can do is evaluating the primary tool

[3] https://catalog.ldc.upenn.edu/LDC2008T19

[4] Annual addresses of U.S. presidents since 1790, available at http://www.presidency.ucsb.edu/sou.php

of these methods: the diachronic word embeddings. Evaluating the quality of word embeddings can be done in a synchronic way, using intrinsic measures. For example, Hamilton et al. (2016) evaluate their word embeddings at the different time periods on a word similarity task using a manually annotated corpus.

In the literature, most authors observe the evolution of some selected words, conducting a qualitative and subjective evaluation, relying on the intuition of known past word meaning change. For example, Kim et al. (2014) observe drifts of words that went through well-known semantic change, such as *gay* and *cell*. They also give interpretations of semantic change, using neighbouring words in the vector spaces of the two periods as well as examples of sentences containing the target word, to check whether the less well-known drift detected by the model makes sense (such as with the word *actually*).

We can also compute the held-out predictive likelihood of the diachronic embeddings model and compare it with a model with embeddings that are stable across time, to evaluate how the temporal aspect of the model improves word representation (Rudolph & Blei, 2018; Bamler & Mandt, 2017; Frermann & Lapata, 2016).

Another possibility is to evaluate diachronic models using a downstream task: predicting the time slice where a word appears. This task is called word epoch disambiguation. It was introduced by Mihalcea and Nastase (2012). They split the Google books corpus into three epochs: 1800±25 years, 1900±25 and 2000±25. For a set of 165 target words, they classify all the occurrences of these words into time slices, using their context and part-of-speech information. Later, the SemEval-2015 Task 7: "Diachronic Text Evaluation" (DTE, Popescu and Strapparava, 2015) took place. A corpus of newspapers published between 1700 and 2010 is divided into slices of 25 years. The task is to identify the time interval when a piece of news was written, among the set of non-overlapping consecutive time slices. Frermann and Lapata (2016) evaluate their model on this task. Lin et al. (2019) use it to evaluate their diachronic approach, classifying documents represented with diachronic embeddings methods. The advantage of this task is that it does not require additional annotation.

It is possible to use external resources to obtain information on the evolution of the senses of words. Dictionaries are a powerful tool, in particular the ones that have published different editions over history. As they are build by committees of lexicographers, they are a particularly reliable resource. The Oxford English Dictionary, for example, is one of the largest dictionaries for the English language. It has several editions, the first one published in 1928. On top of lemma definitions, this dictionary includes the year of first appearance of a sense. Tsakalidis et al. (2019) use it to create a list of 65 words that have gained a new sense in English between the years 2001 and 2013, such as "cloud" and "tablet". A similar dataset is the Word Sense Change Testset (Tahmasebi & Risse, 2017b), a list of 23 words that went through usage or meaning change during the past centuries, together with a list of 11 words that

remained stable. The words were selected using Wikipedia, dictionary.com and the Oxford English Dictionary.

Another possible resource are lexical databases such as WordNet (Fellbaum, 1998). Mitra et al. (2015), on top of using the New Oxford American Dictionary as gold standard to identify senses, use WordNet to perform an automatic evaluation of their semantic change detection method. WordNet is composed of synsets: each synset is a set of lemmas that all have the same meaning, this meaning being one of the senses of a specific word. The authors map each disambiguated sense to senses of synsets in WordNet, using it to confirm that a detected new sense is really distinct from the other senses of the word. Frermann and Lapata (2016) use the same method to validate detected sense births.

However, Tahmasebi et al. (2018) warn against the use of external resources such as WordNet and Wikipedia. The fact that these resources usually lack time information makes them unreliable. They might not cover all vocabulary and word senses of the historical corpora under study; moreover, contrarily to resources such as Oxford English Dictionary, WordNet has no information about when a sense appeared.

3.1.2.2 Annotated corpora

To evaluate diachronic approaches, the type of information one would expect is a human-annotated list of words with a quantified degree of semantic shift for a list of successive time periods. However, such information is very complex to build. A change in word usage is often linked with a slow cultural shift: it is usually very hard to identify the time when the change started, or when a sense took over another one. Thus, all annotated datasets at the time of this thesis only compare the usage of a word between two disjoint time periods.

Until recently, few of such dataset could be found, because of the difficulty of annotation. The first dataset that truly spread in the research community and became a common evaluation tool consists of 100 manually annotated words in English. It was build by Gulordava and Baroni (2011) and is sometimes called the "GEMS" dataset in the literature, following the name of the GEMS Workshop where the associated paper was published in 2011. Recent initiatives from the NLP community start to produce more annotated data, such as the SemEval 2020 Task 1: Unsupervised Lexical Semantic Change Detection (Schlechtweg et al., 2020) and its equivalent in Italy, DIACR-Ita (Basile, Caputo, Caselli, Cassotti, & Varvara, 2020). We can also cite "Diachronic Usage Relatedness" (DURel) in German (Schlechtweg, Schulte im Walde, & Eckmann, 2018). An example of domain-specific annotated dataset is the Reddit corpus (Del Tredici, Fernández, & Boleda, 2019). The authors build a corpus with two non-consecutive time bins, 2011–2013 and 2017, using user-generated language from the r/LiverpoolFC subreddit. The members of

the subreddit annotated 97 words in terms of semantic shift between the two periods, including 34 words with identified semantic shift and the rest being stable confounder words. Among these annotated datasets, we use GEMS, SemEval and DURel for our experiments in this chapter. These three datatsets are described in more details in Section 3.3.1.

The annotation can be made according to two main schemes. The simplest one consists in asking annotators to provide information on each word of the list, without seeing them in context. The second one involves showing words used in sentences from different periods and asking the annotators to rate the relatedness of the usages.

Annotated data for semantic change detection usually consists in lists of words associated with information on their evolution. This information can be of several types. It can be a ranking of the words from most changed to least changed, usually accompanied by scores (as for GEMS, SemEval and DURel). It can be binary information, stating whether each word has undergone semantic change or not. Or it can be pairs of related words, associating a word with an analogous one at a different time period. The latter is called *temporal analogy*.

In this category, Szymanski (2017) proposes to determine the equivalents of a word taken at a given period on other time slices. Such an equivalent exists if it occupies a similar place in the word representation space, for a different time slice. The detection of *temporal analogies* brings a complementary look into the measurement of semantic change: instead of determining words whose meaning has changed over time, we determine meanings whose associated word has changed over time.

Some databases of temporal analogies are made available by Yao et al. (2018) and Szymanski (2017). The former is constructed using articles extracted from the *New York Times* from January 1990 to July 2016. The latter is constructed using the New York Times Annotated Corpus with the help of encyclopedic sources. A database of temporal analogies consists in *[period, word]* pairs, where the word is representative of its concept for the period in question. Since the embeddings of each time slice are all located in the same vector space, they can be directly compared. Thus, solving a temporal analogy of *[period1, word1]* for *period2* consists in determining the vector closest to the one of *word1* at *period1* among the embeddings of *period2*. The word associated with this vector is considered analogous. For example, in the New York Times Corpus, the word "Reagan" designates in 1987 the President of the United States then in office. It is considered equivalent to the word "Bush" in 1990, the name of the president that year.

A limitation of these corpora is the temporal range. Almost all classical semantic change datasets that are human-annotated split the corpus into only two time periods, not allowing us to consider the sequential nature of semantic change. Temporal analogy use several periods, but only provide information about pairs of words at given time slices, without sequential information either.

A solution commonly used in this field to overcome these limitations is to generate a corpus with synthetic semantic drifts.

3.1.3 Generating synthetic semantic drift

As argued by Schlechtweg and Schulte im Walde (2020), annotated test sets for semantic change detection are too small to be able to generalize the results obtained from experimenting on them. One of the main reasons for this small size is the complexity of annotation for this task. It led the research community to turn to the workaround of synthetic data generation. Large synthetic corpora can be generated by creating synthetic drifts over several time slices, overcoming the two-periods limitation of annotated data. Moreover, creating synthetic data allows researchers to control precisely the semantic drifts and all the other parameters involved in language change. Generating a corpus of synthetic semantic change allows us to evaluate models on several points:

- Working on other tasks derived from measuring semantic change, such as binary semantic change (i.e. detecting if a word gained or lost a sense or not), detecting the time period when the change started, and at what speed the change happened.
- Studying the impact of polysemy and frequency, two major variables in semantic change, and how models handle noise in general.
- Distinguishing the category of semantic change undergone by words (did a sense appear, disappear, change...).

An increasing amount of papers rely on the creation of a corpus of synthetic semantic change to tackle the issue of evaluation, with different strategies to build the synthetic corpus. Kulkarni et al. (2015) duplicate a Wikipedia corpus several times to create time slices. They introduce changes in the later time slices by replacing some selected words by an unrelated word according to a chosen probability, and test their models on the detection of the time when the change occurred. The new word used to perturbate the corpus by replacing the old one can be either of the same part-of-speech, or of a different one.

Rosenfeld and Erk (2018) and Shoemark et al. (2019) use pseudo-words: they merge two words that do not share a common sense, creating a pseudo-word, and generate synthetic change by controlling the proportion of sentences using each of the two original words in the successive time slices of a corpus. Dubossarsky et al. (2019) use a similar sense injection system, but with both unrelated and related word pairs, to get better understanding on how much the new sense should be different in order to be detected by semantic change detection methods.

However, as advised by Schlechtweg and Schulte im Walde (2020), it is preferable to use the natural polysemy of words for the synthetic drift to be as

close as possible to reality. Indeed, in language change, the different senses of a word evolve in a coherent way: when a new sense appears, it is usually related to the existing senses of the word. Thus, instead of controlling the proportion of sentences containing two unrelated words merged as a pseudo-word, they advise to use sentences containing different senses of a unique word.

3.2 Preliminary analysis: qualitative comparison of embeddings

In this section, we study in an unsupervised fashion the behaviour of the different kinds of methods introduced in the previous chapters. We apply them to the *New York Times Annotated Corpus* (NYT) introduced in Section 3.1.1, a large corpus of newspaper articles spanning 2 decades. A similar experiment is conducted in Appendix C on the same corpus, focusing on the ability of the different diachronic embeddings models to handle data scarcity.

We study the different methods empirically, by looking at the distribution of the drifts they detect, as well as by comparing them directly through:

1. The correlation between the rankings of the vocabulary by degree of semantic change.
2. The average correlation between the series of drifts for all words.
3. The top drifting words on the full period.

3.2.1 Models and experimental framework

We analyse the behaviour of the different methods introduced in Chapter 2 using BERT contextualised embeddings. We compare them with methods using non-contextual embeddings from Chapter 1. In order to ease the comparison between different non-contextual embeddings methods, we use a common model as a basis: the Dynamic Bernoulli Embeddings model (DBE, Rudolph and Blei, 2018). We implement several variations of this model by changing its regularisation.

3.2.1.1 Dynamic Bernoulli Embeddings and regularisation variants

We implement two variants of the DBE model. The goal is to reproduce the behaviour of different models from the literature, keeping the DBE model as a common basis to get optimal comparability of the results. This also aims at studying more deeply the behaviour of dynamic embeddings and the most useful features to detect semantic change. We rename the classical DBE model as "DBE-Dynamic" to distinguish it from its variants.

In the loss function used to train the DBE model, the last term is a regularisation term on word vectors and on their drift (see Section 1.5.3 for more details):

$$\mathcal{L}_{prior}(\boldsymbol{\rho}, \boldsymbol{\alpha}) = -\frac{\lambda_0}{2} \sum_v \|\boldsymbol{\alpha}_v\|^2 - \frac{\lambda_0}{2} \sum_v \|\boldsymbol{\rho}_v^{(0)}\|^2 - \frac{\lambda}{2} \sum_{v,t} \|\boldsymbol{\rho}_v^{(t)} - \boldsymbol{\rho}_v^{(t-1)}\|^2. \quad (3.1)$$

The regularisation of the classical DBE aims at preventing the vectors at period t from going too far apart from those at $t-1$. We define a first variant, which goes closer to the idea of incremental updating (Kim et al., 2014). We remove the constrain on the temporal drift of the embeddings in the DBE model. In that situation, the loss function (Equation 1.10) only includes the first two terms of the log-prior (Equation 3.1) on top of $\mathcal{L}_{pos}$ and $\mathcal{L}_{neg}$. We name this variant DBE-Incremental.

The second variant consists in deleting the hypothesis of chronology in the successive temporal word vectors. The last term of the log-prior $\mathcal{L}_{prior}$ is replaced by $\sum_{v,t} \|\boldsymbol{\rho}_v^{(t)} - \boldsymbol{\rho}_v^{(0)}\|^2$. It forcefully prevents the vector $\boldsymbol{\rho}_v^{(t)}$ to grow too far from its original position $\boldsymbol{\rho}_v^{(0)}$, which is the prior of the representation of the first time period $\boldsymbol{\rho}_v^{(1)}$. This idea is similar to the model of Han, Gill, Spirling, and Cho (2018) or Hamilton et al. (2016) where the diachronic embeddings are learned independently on each time slice. This variant is named DBE-Independant.

3.2.1.2 Experimental framework

We apply these different methods to the *New York Times Annotated Corpus*[5] (NYT, Sandhaus, 2008). It collects around 1 855 000 articles from the New York Times ranging from January 1st 1987 to June 19th 2007. We divide the corpus into $T = 20$ yearly time steps (the incomplete last year is not used in the analysis).

For the DBE models, we remove stopwords and least frequent words. To tune the hyperparameters, we use the log-likelihood of positive examples $\mathcal{L}_{pos}$ measured on the validation set (10% of the full corpus). We train each model for 100 epochs, with a learning rate of 0.1, using the Adam optimiser. We choose an embedding dimension $d = 100$. The initial drift parameter λ_0 is set to $\frac{\lambda}{1000}$ as advised by Rudolph and Blei (2018), and λ is set to 1. Finally, we use a context window of 4 words and a negative ratio of 10. The embeddings are trained with 1000 mini-batches per time step. A static model is first trained on the whole dataset. Its parameters can then be used to initialise the dynamic model's $\boldsymbol{\rho}$ and $\boldsymbol{\alpha}$.

We want to observe the global behaviour of the methods, thus requiring to analyse a large vocabulary instead of a limited list of hand-picked target words.

[5] https://catalog.ldc.upenn.edu/LDC2008T19

We select the top 10k most frequent words (excluding stop-words) to quantify their semantic change over time. For the DBE models (DBE-dynamic, DBE-incremental and DBE-independant), to measure semantic change of words, we use the cosine distance.

For the BERT models, we extract the contextualised embeddings from the corpus using the scalable method introduced in Section 2.3, leading to 200 embeddings per word for each of the 20 time slices. The scalable method is necessary given the large number of words in the vocabulary and the size of the corpus. We use the pre-trained bert-base-uncased model from the `transformers`[6] library. Then, we apply the different aggregation methods introduced in Chapter 2: the Average Pairwise distance (APD); the averaging method with cosine distance to compare average embeddings between time slices (Avg); the clustering method (k-means with $k = 5$), comparing clusters distributions with the Jensen-Shannon divergence (Clust-JSD) or comparing average embeddings inside clusters with the Wasserstein distance (Clust-WD); and the Wasserstein distance on all 200 embeddings (WD).

3.2.2 Comparison

We compare the distribution of drifts and the most drifting words according to each method. We also compute the correlations between the rankings and the drifts time series. It allows us to conclude on the respective limitations of all methods.

3.2.2.1 Comparison of drifts distribution

In this section, we examine the distribution of word drifts outputted by each model. We plot the superimposed histograms of successive *inceptive* drifts from $t_0 = 1987$ to each subsequent time step, for all studied models, for the 10k words under study. On the histograms, the darkest colour curve represents the drift between $t_0 = 1987$ and $t = 1988$ and the lightest one, the drift between $t_0 = 1987$ and $t = 2006$. Figure 3.1 shows how the histograms are build in more details. Note that the figures of this section require colours to be interpreted.

Figure 3.2 shows these drift histograms for each model under study. The first 5 plots are BERT embeddings with the different aggregation methods and distance measures. The last 3 plots are the variants of the DBE model with drift computed using cosine distance.

Highlighting directed drifts. A first crucial property is the *directed* aspect of the drifts: when the word vectors progressively drift away from their initial representation in a directed fashion. This behaviour can be identified on the plots by the color gradient. The classical DBE model (DBE-dynamic)

[6] https://huggingface.co/transformers/

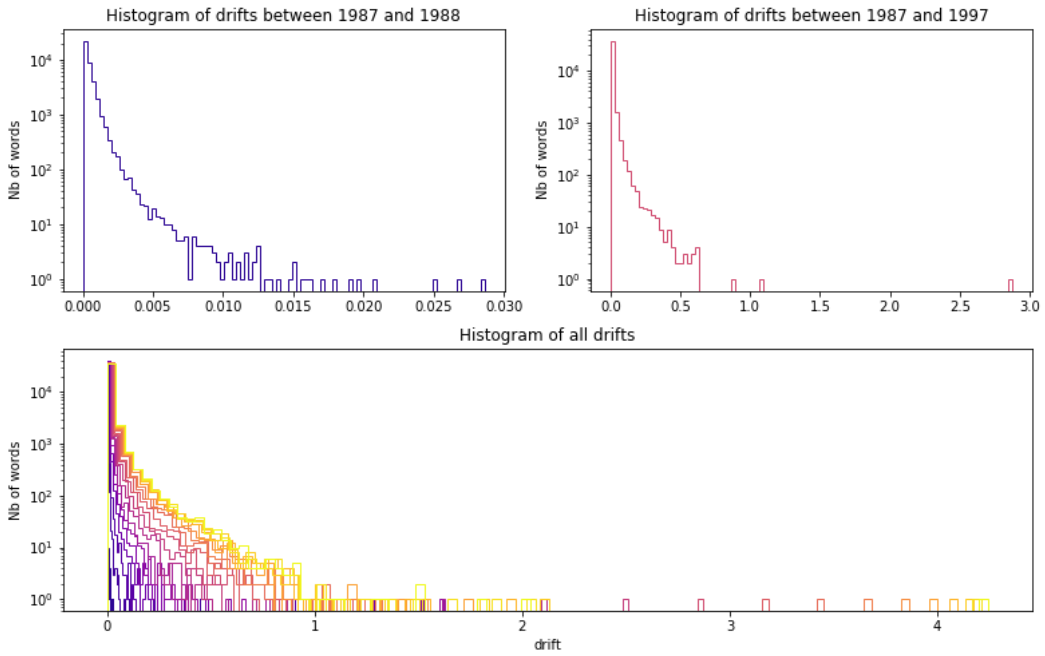


Figure 3.1 – Histogram of drifts between the embeddings at $t_0 = 1987$ and t for various values of t with the DBE model. Two years are singled out to expose how the full histogram is built, note that the abscissa scale differ. The ordinate axis is logarithmic.

shows this behaviour well, with a very clear colour gradient. This means that the model mainly captures drifts with a trend, rather than short changes in embeddings followed by returns to normal. This is explained by the use of the diffusion process which link the time steps in equation 1.9: it allows the dynamic model to emphasise the directed nature of drifts and to ignore brief variations. These brief variations are due to events that temporarily modify the context in which a word appears without having a long-term impact on its meaning. They are captured by the DBE-independant version of the model, whose histogram does not show a directed evolution of drift as a function of distance to t_0, and therefore does not distinguish these "noises" from the general trend in the evolution of words. Finally, despite the absence of a regularisation term on drift, the DBE-incremental model naturally captures a relatively directed drift over time although the histogram shows a greater sensitivity to noise than the classical DBE model.

Among the methods using BERT embeddings, only APD does not show directed drifts. It makes sense, as the main idea behind APD is to compare all occurrences of a word in a period to all the ones of another period: there is no process to smooth the outliers and the noise or to sum up the main information. On the contrary, methods based on averaging or clustering of BERT embeddings manage to highlight the directed nature of word usage drift. Orevall, no color gradient is as "clean" as the one from the DBE-dynamic model. However, word usage change is never perfectly directed; by enforcing

this property to this extend through drift regularisation, the model can create spurious directed drifts or hide important jumps.

Discriminating stable words. The second property to highlight is the capacity of the models to discriminate words that drift from words that stay stable. From the human point of view, a majority of words has a stable meaning (Gulordava & Baroni, 2011); especially on a dataset covering only two decades like the NYT. On the histograms, this characteristic is revealed by the distribution of the yellow curve (drift between the first time slice and the last one): when it takes low values for a lot of words, it means that most words are stable. The DBE-dynamic and DBE-incremental histograms have a large number of words with very low drift. The BERT-averaging and BERT-clustering histograms have a very similar shape. The DBE-independant model, by introducing a regularisation with respect to the initial embeddings, strongly enforces this property: a large part of the words are almost invariant over the whole corpus, and only a selection of drifts stand out.

A system a bit different is BERT-WD, which takes the Wasserstein distance between the full set of 200 embeddings per period. The amount of drifts that the words go through is globally higher, with a peak further from zero than for the other systems. Finally, the APD curve has a very different distribution shape, with the peak being close to the middle of the distribution. It resemble a normal distribution (plotted on a logarithmic axis), while one would expect the drifts to have an exponential distribution with few words having a large drift and many words having a low one.

3.2.2.2 Correlation between drifts

We compare the different systems together using correlation, to determine whether they capture the same information.

First, we compare the word drift ranking outputted by each model. We quantify the semantic change of each of the 10k target words by computing the distance between the first and the last time slice (the *full drift*). Then, we rank the words by distance and use the Spearman rank correlation to make the comparison. The results can be found in Table 3.1. As expected, the models using the same information have high correlation. In particular, the two clustering methods, which compare distributions of clusters, are highly correlated. The averaging, APD, and the two methods using the Wasserstein distance, all compare BERT embedding vectors and are correlated. However, almost none is correlated with the rankings of the DBE models, meaning that they do not capture the same information. Similarly, the ranking from clustering with JSD is completely uncorrelated with the APD ranking. To better understand this behaviour, we study another kind of correlation.

The second experiment compares the sequential drifts of the words. We use

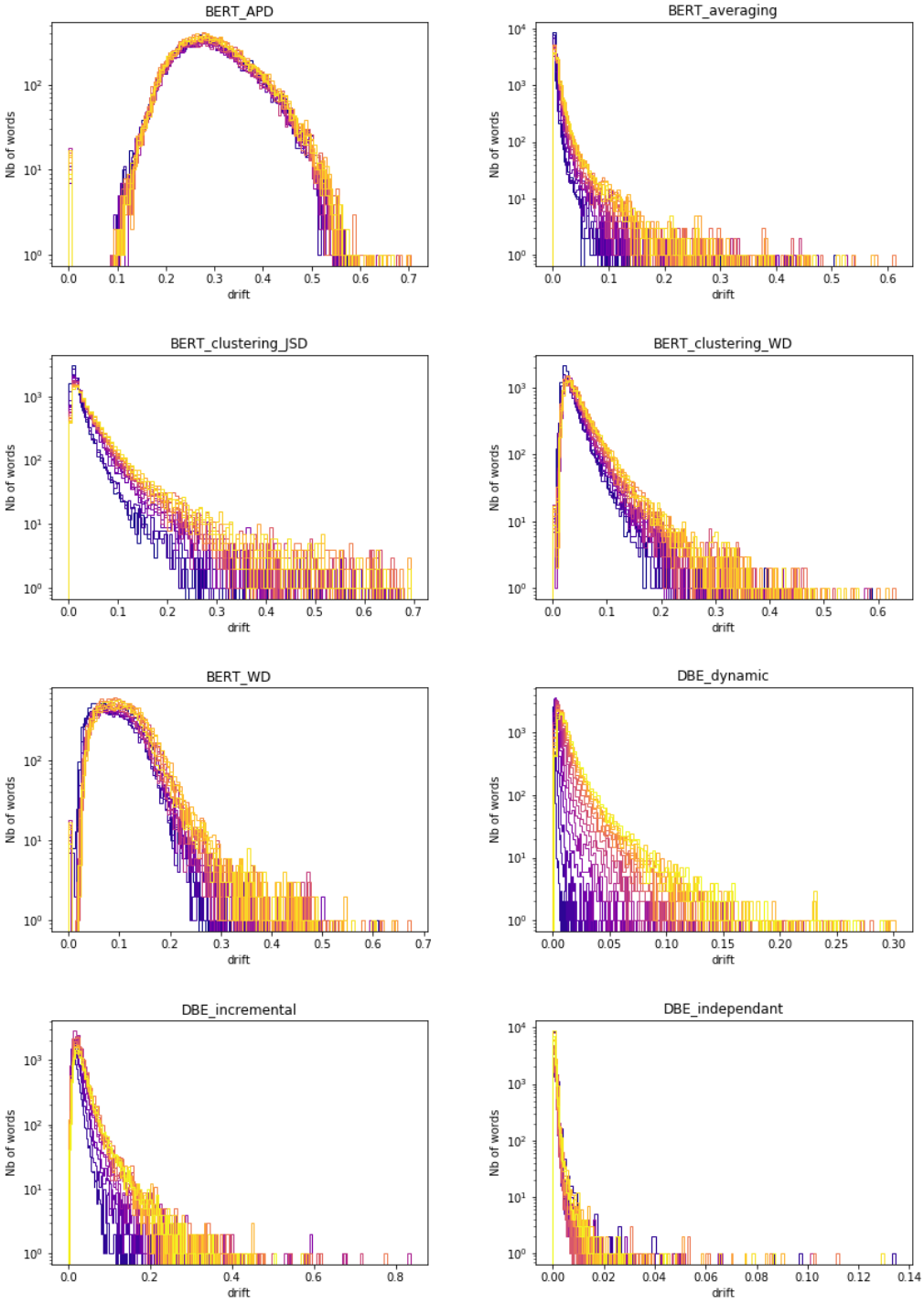


Figure 3.2 – Histogram of word drift for each model on the NYT corpus. The drifts are computed from $t_0 = 1987$ to each successive time step, and superposed on the histogram. The lightest colours indicate drifts calculated until the most recent time steps. The number of words are on logarithmic scale.

	BERT					DBE		
	APD	Clust_JSD	Avg	Clus_WD	WD	Indep	Incr	Dyn
APD	1.000							
Clust_JSD	0.001	1.000						
Avg	0.354	0.664	1.000					
Clust_WD	0.607	0.667	0.776	1.000				
WD	0.460	0.385	0.691	0.555	1.000			
DBE_Indep	0.005	-0.041	-0.048	-0.025	-0.101	1.000		
DBE_Incr	-0.018	0.040	0.056	0.016	0.117	0.283	1.000	
DBE_Dyn	0.001	0.005	0.014	0.006	0.001	0.342	0.754	1.000

Table 3.1 – Spearman Rank Correlation between system output rankings. Grey values indicate non-significant correlation (p-value > 0.05).

the time series of drifts computed for the histograms: the distance between the embeddings at the first time slice and at each successive time slice (*inceptive* drift). Then, for each word, we compute the Pearson correlation between its drift time series in each model. We average the correlations for all words.

The results can be found in Table 3.2. This time, the DBE models and some BERT systems are correlated; in particular, the averaging method has the highest correlations with the DBE models' drifts. This is justified by the behaviour of the averaging method, which outputs a unique embedding per time slice exactly as the DBE models work. The APD drifts are much less correlated with the other models, except for the Wasserstein-based systems. This is confirmed by the shapes of their respective histograms, which are relatively similar.

	BERT					DBE		
	APD	Clust_JSD	Avg	Clus_WD	WD	Indep	Incr	Dyn
APD	1.000							
Clust_JSD	0.184	1.000						
Avg	0.175	0.490	1.000					
Clust_WD	0.335	0.769	0.576	1.000				
WD	0.426	0.332	0.695	0.437	1.000			
DBE_Indep	-0.026	0.054	0.112	0.059	0.040	1.000		
DBE_Incr	-0.022	0.175	0.348	0.194	0.199	0.358	1.000	
DBE_Dyn	0.031	0.278	0.507	0.303	0.352	0.250	0.600	1.000

Table 3.2 – Average Pearson correlation between the drift time series of all words.

3.2.2.3 Comparing top drifting words

We compare the list of most drifting words of each model. The words that differ between models reveal the specificities and limitations of these models.

Tables 3.4 and 3.3 show the top 20 most drifting words for all models. Furthermore, among the top 100 words of all models, BERT and DBE all together, we select the ones that are common to more than 4 models and highlight them in blue in the tables. Thus, in the BERT table (Table 3.4), some highlighted words are not necessarily in the top 100 of any DBE model. Conversely, all highlighted words in the DBE table (Table 3.3) appear in the top 100 of at least one BERT-based system.

In the top drifting words of all models, we find a lot of words in common. Among them, we mostly find name entities such as "Bloomberg" and "Katrina." Another word high in the ranking for all models is "web," closely followed by "net." These words refer to events that had a large media coverage and resonance (e.g hurricane Katrina), or technological trends that flourished during the period under study. On top of this, we can highlight several differences between the rankings.

First, we can find in the top drifting words of all BERT models, examples of words that have different possible part-of-speech. For example, "lead" and "share" are respectively in the top 20 and 50 for all BERT models but below 1000th rank for DBE models. This is due to the language modeling of BERT being sensitive to syntactic information, thus passing on syntactic variation when computing distance measures.

Second, some words with very diversified context such as function words showcase large drifts according to the methods that do not use a unique embedding per time slice (all systems except DBE models and Averaging method). This is especially visible with APD ranking, with words such as "base," "cross" or "stand"; all words that can have very diversified contexts. This reflects high intra-period diversity more than actual semantic change. Overall, APD is the system with the most different list of top drifting words compared to the other BERT systems. For example, two of the words with the largest APD for the full period are "cross" and "post," respectively ranked 5 and 7. However, they are ranked below 400 for Full-WD and even below 1000 for all other systems.

Among other BERT systems, the top words from the Clustering + JSD method also diverge a bit compared to the other BERT methods. This might be due to its distribution tail which is much heavier than the tail of the other distributions, harming the ability to identify extreme drifts (see Figure 3.2). It can be explained by the low dimension of the representation (here, $k = 5$ clusters) that limits the expressiveness of the distance function between clusters distributions of two periods. An example of difference in ranking is "Bloomberg," which is ranked 18th by the Clust-JSD system but much lower by other BERT systems. It has two distinct senses; first, it can refer to Michael Rubens Bloomberg, American businessman and mayor of New York City from 2002 to 2013. Second, it corresponds to the media and financial company Bloomberg L.P co-founded by him. The Clustering + JSD system probably ranks it high thanks to the simplification of the information into small cluster distributions, focusing on distinguishing the two senses rather than measuring

the difference between the embeddings. Note that this word is ranked high by the DBE systems.

DBE dynamic and incremental models show many top drifting words that do not appear for the other methods, such as "Spitzer" and "Contras." These words correspond to entities or events with low long-term echo in the news, especially visible through unbalanced frequency across the time slices. The incremental aspect of the two models keeps the information across the subsequent periods, but it is not visible when computing the full drift (using only the first and the last time periods) for the BERT models.

On the contrary, two types of words are ranked high on the lists of top drifting words for all models. First, words that are associated with events spanning a long period such as "Enron." Second, events with long-term echo such as "Katrina." This is not always the case, or to a lesser extent, for the method with APD measure; for example, the ranking of Katrina is only 209 and the ranking of Enron is 8355.

DBE_independant	DBE_incremental	DBE_dynamic
katrina	katrina	bloomberg
hurricane	corzine	katrina
rumsfeld	esthetic	contras
enron	texaco	nyt
lay	bloomberg	meese
corzine	lieut	rumsfeld
iran	spitzer	corzine
iraq	nyt	enron
uranium	enron	shultz
chad	contras	gorbachev
compensation	qtr	web
gossip	muslims	texaco
sudan	otc	dukakis
bonds	outst	lead
shiite	rumsfeld	ensure
leak	hezbollah	nicaraguan
immigrants	shultz	spitzer
orleans	brantley	soviet
duke	euro	reagan
number	bork	nicaragua

Table 3.3 – List of words with highest full drift (from first to last time slice) for each DBE model on the NYT corpus. Highlighted cells have words that are common to the 100 most drifting words for more than 4 models.

3.3 Experiments on labeled corpora

In this section, we describe the annotated data and the corpora used for the evaluation. Then, we explain the experimental details, in particular the

APD	Clust_JSD	Avg	Clust_WD	WD
laden	treas	laden	laden	laden
sec	ers	nhl	nhl	earns
lay	nyse	earns	earns	nhl
lengths	nba	web	web	web
cross	katrina	rev	lead	sec
lead	earns	homeland	bin	rev
post	officiated	katrina	sec	bin
bin	condolences	lead	share	net
doubles	brantley	bin	homeland	lead
web	web	net	net	homeland
stem	amex	inc	katrina	katrina
earns	texaco	nyse	rev	share
hip	misidentified	share	shorter	stem
pan	nyt	reports	santa	dell
passes	nhl	amex	dell	cent
bears	gundy	sec	hip	inc
wake	lead	amazon	lay	handicap
net	bloomberg	stem	passes	euro
base	noriega	revenue	amazon	amex
sets	net	euro	doubles	nee

Table 3.4 – List of words with highest full drift (from first to last time slice) for scalable BERT embeddings with different aggregation methods and semantic change measures, on the NYT corpus. Highlighted cells have words that are common to the 100 most drifting words for more than 4 models.

pre-trained models used and the fine-tuning process. We develop a brief study on the choice of the number of epochs for BERT fine-tuning.

3.3.1 Annotated data description

We evaluate the proposed semantic change detection methods using six existing manually annotated datasets. An overview of these corpora can be found in Table ??.

GEMS on COHA The first dataset is GEMS, built by Gulordava and Baroni (2011). It consists of 100 words from various frequency ranges, labelled by five annotators according to the level of semantic change between the 1960s and the 1990s. They use a 4-points scale from "0: no change" to "3: significant change", the inter-rater agreement was 0.51 ($p < 0.01$, average of pair-wise Pearson correlations). The most significantly changed words from the dataset are, for example, *user* and *domain*; on the other hand, words for which the meaning remain intact are for example *justice* and *chemistry*. Gulordava and Baroni (2011) describe the procedure used to build the dataset: the annotators ranked words using their intuition, without looking at the context. This procedure is not optimal, since an annotator may forget or not be aware of

some senses of a word (especially its senses in the 1960s).

Still, this dataset is a valuable resource and has been used to evaluate methods for measuring semantic change in previous research (Frermann & Lapata, 2016; Giulianelli et al., 2020). Following previous work, we use the average of the human annotations as semantic change score. For evaluation, we compute Spearman rank correlations between this score and a model output.

To extract contextualised embeddings, we use the Corpus of Historical American English (COHA) [7] introduced in Section 3.1.1. We focus our experiments on the most recent data in this corpus, from the 1960s to the 1990s (with 2.8 million words for the 1960s period and 3.3 million for the 1990s one), to match the manually annotated data. The fine-tuning of the model is also done only on this subset.

Semeval The recent SemEval-2020 Task 1 Unsupervised Lexical Semantic Change Detection (Schlechtweg et al., 2020) involved the construction of 4 annotated datasets in different languages: German, English, Latin and Swedish. All SemEval-2020 Task 1 corpora contain only two periods. The organizers employed an approach to build the annotated corpora that differ from the GEMS dataset. The annotators had to label the relatedness of word senses in a pair of sentences drawn from the two time periods. This system does not require the annotators to precisely identify a sense for each occurrence of a word, a difficult task when dealing with the delicate problem of slight nuances in words usage evolution. In total on the 4 languages, approximately 100k sentence pairs were annotated, with an average inter-annotators agreement score of 0.62. Then, the lexical semantic change score of a word is defined as the Jensen-Shannon Divergence (Lin, 1991) between the sense distributions at the two time periods.

We generate contextualised embeddings for all target words on the four corpora provided by the organizers of the task, English (37 targets, about 13.4M words in the corpus), German (48 targets and 142M words), Swedish (40 targets and 182M words) and Latin (31 targets and 11.2M words). Note that contrarily to COHA, the sentences in the corpora are shuffled and lemmatized.

The challenge defines two subtasks: Subtask 1 is binary classification, i.e. to determine whether a word has changed or not; SubTask 2 aims at ranking a set of target words according to their rate of semantic change. We evaluate our methods on Subtask 2, but propose some solution for Subtask 1 in Section 3.5.

DuReL on DTA The DURel dataset (Schlechtweg et al., 2018) is composed of 22 German words, ranked by semantic change by five annotators between two time periods, 1750–1799 and 1850–1899. Similarly to SemEval, the ranking was build by evaluating the relatedness of pairs of sentences from

[7] https://www.english-corpora.org/coha/

Dataset	GEMS	SemEval				DURel
Language	English	English	German	Swedish	Latin	German
Corpus	COHA	CCOHA	DTA+BZ+ND	Kubhist	LatinISE	DTA
Nb of tokens	6.1M	13.4M	142.5M	182M	11.2M	63M
Period 1	1960–1969	1810–1860	1800–1899	1790–1830	−200–0	1750–1799
Period 2	1990–1999	1960–2010	1946–1990	1895–1903	0–2000	1850–1899
Nb of targets	100	37	48	40	31	22
Agreement	0.51	0.69	0.59	0.58	/	0.66

Table 3.5 – Overview of the 6 manually annotated datasets.

two periods, on a 4-points scale ranging from "identical" to "unrelated". The inter-annotator agreement is high, with an average pairwise correlation of 0.66.

We generate embeddings for the target words using the DTA corpus in German (the 1750–1799 period has about 25M tokens and 1850–1899 about 38M)[8]. The sentences are shuffled, similarly to the SemEval corpora.

3.3.2 Experimental details

We compare the various approaches described in Chapter 2 to detect semantic change. In our experiments we use pre-trained BERT models from the TRANSFORMERS library[9]. They all have 12 attention layers and hidden layers of size 768.

For the COHA corpus, we use the bert-base-uncased model, which was pretrained on the BookCorpus (Zhu et al., 2015) — a dataset of books with 800M word — and English Wikipedia (2 500M words).

For the SemEval Task 1 evaluation set, we fine-tune the BERT models and generate contextualised embeddings on the four corpora provided by the organizers of the task. We use specific models for each language—for English: bert-base-uncased model (same as for COHA), for Swedish: bert-base-swedish-uncased (https://github.com/af-ai-center/SweBERT), for German: bert-base-german-cased (https://deepset.ai/german-bert), for Latin: bert-base-multilingual-uncased model. German is the only language for which we use a cased model since most target words are nouns, which are capitalized in German. The only model available for Latin is a multilingual BERT model trained on 104 languages, including Latin. The rationale behind this choice lies in the fact that many words from some of these languages (e.g., Italian, Spanish, French, ...) have Latin origins.

For the DTA corpus, we generate embeddings using bert-base-german-cased.

For each language, the model is fine-tuned for five epochs on the associated

[8] https://www.ims.uni-stuttgart.de/en/research/resources/experiment-data/durel/
[9] https://huggingface.co/transformers

corpus for domain adaptation. This fine-tuning is unsupervised, i.e. a masked language model objective is used in the fine-tuning step (Devlin et al., 2019) in order to adapt each model to a specific corpus. The model is fed one sentence at a time for SemEval and DTA corpora, since the sentences in these corpora are shuffled. In the case of COHA, the corpus is divided into sets of several sentences until a limit of 256 tokens. These sentences are tokenized by using pretrained BERT tokenizers, which rely on a byte-pair encoding-like scheme (Wu et al., 2016). If sentences are longer than 256 tokens, they are truncated. No additional preprocessing is conducted. Note that the SemEval corpora have already been lemmatized and all the punctuation have been removed, which is a disadvantage for language models such as BERT pre-trained on raw data.

The fine-tuned models are used to generate token embeddings. The corpus for each language is split into two periods and the fine-tuned models are fed with sentences containing one or more target words from the sub-corpus. The sentences are split into tokens, and an embedding of dimension 768 is generated for each token by summing the last four encoder output layers of BERT, as described in Section 2.2.

Note that in some cases, the BERT tokenizer generates tokens that correspond to sub-parts of words. To generate embedding representations for the target words split into sub-parts, we average the embeddings of each subword token constituting a word. For example, in the list of hundred target words for evaluation in the COHA corpus, *sulphate* is divided into two subwords *sul* and *##phate*, where *##* denotes the splitting of the word.

Finally, we store embeddings according to the *standard* method and the *scalable* methods. In the first case, we obtain a contextual embedding representation for each target-word usage, together with the time period it belongs to. In the second case, we store only 200 embeddings per time slice, by clustering and averaging them as a stream.

3.3.3 Impact of fine-tuning

Using the COHA corpus, we fine-tune the pre-trained bert-base-uncased model for 1 to 10 epochs to study the impact of fine-tuning on the results.

Figure 3.3 shows the influence of fine-tuning for two clustering methods (affinity propagation, and k-means with $k = 5$). A light fine-tuning (for one epoch) decreases the performance of both methods (in terms of Spearman correlation with ground truth ranking) in comparison to no fine-tuning at all (zero epochs). After that, the number of fine-tuning epochs until up to 5 epochs is linearly correlated with the performance increase. Fine-tuning the model for five epochs appears optimal. With larger number of epochs, the performance for both methods starts decreasing, most likely because of over-fitting due to the reduced size of the fine-tuning dataset compared to the training data.

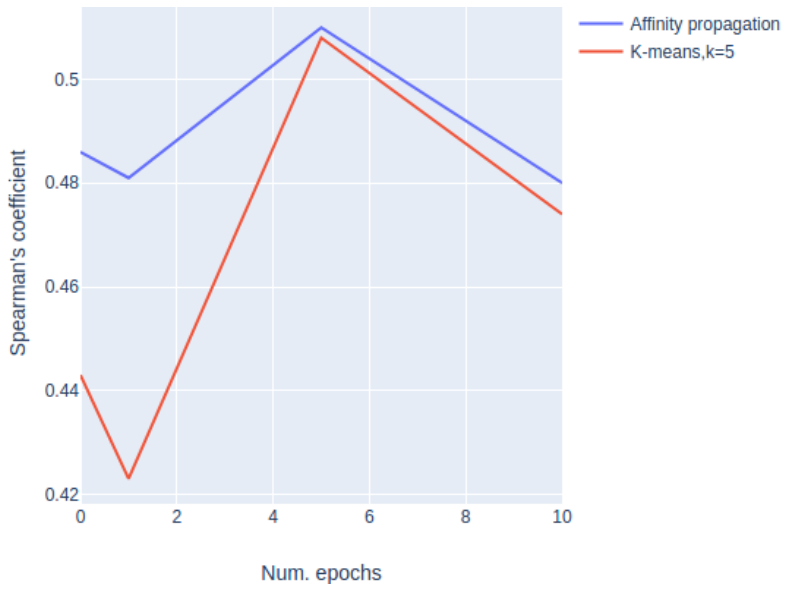


Figure 3.3 – Impact of BERT fine-tuning on the performance of two distinct aggregation methods, affinity propagation and k-means with $k = 5$, for the GEMS dataset on the COHA corpus.

Table 3.6 shows Spearman correlations between gold standard and the measured semantic change using affinity propagation clustering and JSD, on embeddings extracted from pre-trained and fine-tuned BERT models. Fine-tuning BERT improves the accuracy on average, and the impact is most beneficial to English.

	GEMS	SemEval				DURel	Avg
		English	Latin	German	Swedish		
Pre-trained BERT	0.486	0.216	0.481	0.488	-0.072	0.512	0.352
Fine-tuned BERT	0.510	0.313	0.467	0.436	-0.026	0.542	0.374

Table 3.6 – Spearman Rank Correlation between system output rankings and ground truth rankings for all datasets. We use standard embeddings with affinity propagation clustering and JSD to measure semantic change. Grey values indicate non-significant correlation (p-value > 0.05).

3.4 Detailed analysis

In this section, we consider different experiments to deepen the analysis:

1. A comparison of the two extraction methods (standard and scalable embeddings) with different aggregation methods and distance measures.
2. A comparison of the different solutions relying on optimal transport:

	GEMS	SemEval				DURel	Avg
		English	Latin	German	Swedish		
STANDARD EMBEDDINGS							
Averaging	0.349	0.315	**0.496**	**0.565**	0.212	0.656	**0.432**
k-means 5 JSD	0.508	0.189	0.324	0.528	0.238	0.560	0.391
aff-prop JSD	**0.510**	0.313	0.467	0.436	-0.026	0.542	0.374
APD Cos	0.176	**0.514**	0.315	0.262	0.150	-0.035	0.230
APD Canb	0.314	0.486	0.397	0.163	0.290	-0.151	0.250
SCALABLE EMBEDDINGS							
k-means 5 JSD	0.430	0.316	0.358	0.508	0.073	**0.658**	0.390
aff-prop JSD	0.394	0.371	0.346	0.498	0.012	0.512	0.355
APD Cos	0.268	0.504	0.324	0.287	0.219	0.249	0.308
APD Canb	0.328	0.483	0.304	0.479	0.226	0.288	0.351

Table 3.7 – Spearman Rank Correlation between system output rankings and ground truth rankings for all datasets. Grey values indicate non-significant correlation (p-value > 0.05).

whether to do one or multiple clusterings, to use Sinkhorn regularisation or not, to use weighted averaging of embeddings inside clusters or not.

3. An evaluation of the effect of the clusters merging method for scalable embeddings.

4. A summary of the our methods, compared with other methods from the literature.

5. An error analysis, to show examples of spurious drift detected by our methods and explain them. In addition, we propose a method to tackle some of the errors detected before, through filtering of the clusters.

3.4.1 Standard VS scalable extraction

We apply k-means and affinity propagation to the 6 evaluation corpora. We make use of the Scikit-learn implementation for both algorithms[10]. For k-means, we set the number of clusters $k = 5$ and use default parameters for the rest. Similarly, for affinity propagation, we use the default parameters set by the library.

We use the different methods introduced in Chapter 2 to measure semantic change on embeddings extracted using the standard methods (one vector for each occurrence of a target word) and the scalable method (only 200 vectors for each target word at each period). The results can be found in Table 3.7.

[10] https://scikit-learn.org/stable/modules/clustering.html

The best performing method differs for the different datasets. However, on average, the averaging method works best. The methods that perform worse are the ones using average pairwise distance. The APD results are better with Canberra distance (APD Canb, see Section 2.2.2) compared to Cosine distance (APD Cos) but the rankings it outputs fail to correlate with the gold truth ranking for several test corpora.

K-means performs better than affinity propagation on average. The difference in performance between k-means and affinity propagation could be partially explained by the different number of clusters in the two approaches. Affinity propagation outputs a huge amount of clusters, 160 on average on the GEMS dataset. The particular number of clusters found by affinity propagation for a word correlates strongly with the frequency of that word in the corpus with correlation coefficient $r = 0.875$, as is illustrated in Figure 3.4. For instance, the word *woman* which occurred over 20k times in both time slices in COHA has the most number of clusters, 972, while *negligence*, occurring just 76 times has the least amount of clusters, 10. Thus, determining the optimal number of clusters for different words is not straightforward. Affinity propagation usually produces a skewed clustering, with a large number of small clusters containing only one or two data points.

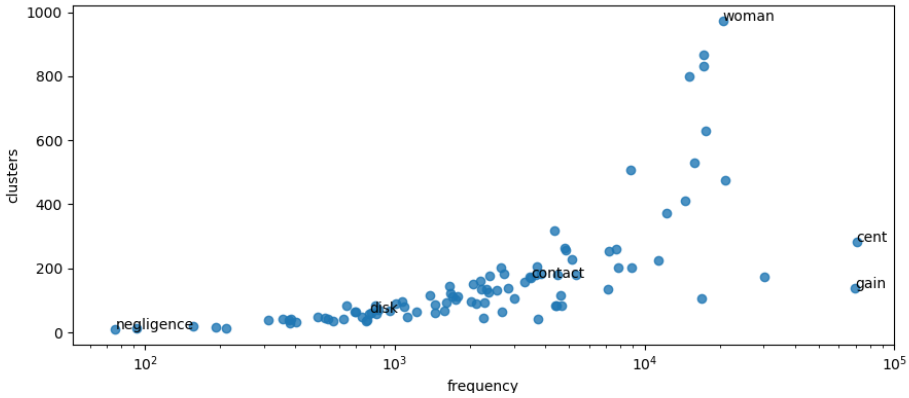


Figure 3.4 – Number of clusters found by affinity propagation on standard embeddings and frequency of words in the 1960s and 1990s in COHA for all GEMS target words.

For the clustering methods, the scalable embeddings extraction performs comparably for k-means and slightly worse for affinity propagation, compared to the standard embeddings. But this hides high disparities between evaluation corpora. As for the APD method, it performs better on scalable embeddings compared to standard embedding. It makes sense, as a lot of the noise from the standard embeddings is removed when applying the merging algorithm to obtain only 200 embeddings. Moreover, the computation time is much shorter with scalable embeddings. It is especially valuable for the APD and the affinity propagation algorithm which can be very time-consuming with a large number of points.

3.4.2 Optimal transport

We use the different methods relying on optimal transport (OT) to compute the distance between embeddings at two time periods, for both standard and scalable embeddings. We compare it with the Jensen-Shannon divergence (JSD) which measures the divergence between cluster distributions, and the averaging method which led to the best performance in the previous section.

As a reminder, the OT methods involve computing the average embedding inside each cluster in each period, and calculating the Wasserstein distance (WD) between these average embeddings weighted by the clusters distribution of each period. We can either do a unique clustering on the embeddings of the two periods, or independent clustering at each time slice (denoted "2×" in the results table). Moreover, in the case of the scalable embeddings, each vector of the 200 embeddings is associated with the number of vectors which were averaged to create it; this can be used as distribution to compute the WD without performing clustering. These counts can also be used as weights when computing the average inside each cluster; this is denoted "WD-weighted" in the results table. Finally, a regularised version of the WD can be computed, the Sinkhorn distance. The parameter λ controls this regularisation. We compute the Sinkhorn distance with $\lambda \in \{10^{-2},\ 10^{-1},\ 1,\ 10\}$. More details on the Wasserstein distance and Sinkhorn regularisation can be found in Section 2.2.5.

Table 3.8 shows a selection of results. The results from all systems combinations can be found in appendix B, in tables B.2 and B.1.

The WD tends to give better results than JSD; it is especially the case on corpora where averaging outperforms clustering. Indeed, the OT system uses information coming from both the cluster distribution and the embeddings vectors. An extreme example is the Swedish SemEval dataset, where the clustering with JSD performs particularly poorly: using the WD, which takes into account the average embeddings on top of cluster distributions, greatly increases the correlation with the gold standard. On the contrary, on COHA where averaging performs poorly in comparison to clustering, WD is underperforming.

Performing two independent clusterings does not improve the results compared to a unique clustering, on average. Big gaps can be observed in both directions for some test corpora. However, as the number of target word is small, it could be explained by only a few words benefiting or suffering from the additional degree of freedom given by the independent clusterings. Indeed, on the biggest dataset GEMS (100 words), the performances of unique and independent clusterings are comparable. We deduce that computing the intra-cluster average per time period already gives enough information to compare the embeddings between periods. Similarly, using vector counts as weights when averaging the embeddings inside the clusters does not improve the accuracy.

Clustering	Measure	GEMS	SemEval				DURel	Avg
			English	German	Swedish	Latin		
STANDARD EMBEDDINGS								
averaging	CD	0.349	0.315	**0.565**	0.212	0.496	0.656	**0.432**
aff-prop	JSD	**0.510**	0.313	0.436	-0.026	0.467	0.542	0.374
aff-prop	WD	0.386	**0.643**	0.490	-0.100	0.246	0.456	0.354
2× aff-prop	WD	0.392	0.419	0.377	0.233	0.327	0.379	0.355
2× aff-prop	sinkhorn 0.01	0.404	0.435	0.376	0.250	0.296	0.307	0.345
k-means 5	JSD	0.508	0.189	0.528	0.238	0.324	0.560	0.391
k-means 5	WD	0.334	0.347	0.408	0.248	0.470	0.618	0.404
2× k-means 5	WD	0.337	0.409	0.432	0.290	0.457	0.514	0.407
SCALABLE EMBEDDINGS								
aff-prop	JSD	0.394	0.371	0.498	0.012	0.346	0.512	0.355
aff-prop	WD	0.369	0.456	0.421	0.264	0.397	0.484	0.399
2× aff-prop	WD	0.380	0.412	0.457	0.190	0.426	0.530	0.399
2× aff-prop	WD-weighted	0.358	0.419	0.463	0.214	0.395	0.551	0.400
k-means 5	JSD	0.430	0.316	0.508	0.073	0.358	**0.658**	0.390
k-means 5	WD	0.372	0.360	0.514	0.316	0.360	0.607	0.430
k-means 5	sinkhorn 10	0.218	0.532	0.432	0.265	0.465	0.468	0.397
2× k-means 5	WD	0.339	0.375	0.482	0.299	**0.502**	0.512	0.418
2× k-means 5	sinkhorn 0.1	0.331	0.445	0.484	0.257	0.484	0.549	0.425
2× k-means 5	WD-weighted	0.323	0.368	0.465	0.302	0.490	0.518	0.411
none	WD	0.312	0.386	0.416	0.252	0.283	0.526	0.363
none	sinkhorn 0.01	0.316	0.393	0.393	0.263	0.274	0.567	0.368

Table 3.8 – Spearman Rank Correlation between system output rankings and ground truth rankings for all datasets. Grey values indicate non-significant correlation (p-value > 0.05).

Comparing the full list of 200 embeddings without doing clustering leads to slightly worse performance on average. This confirms the conclusions drawns from the low performance of the APD method: aggregation is necessary to avoid being too sensitive to noise. Furthermore, the OT methods on scalable embeddings always lead to better performance on average compared to the standard embeddings.

Finally, the Sinkhorn regularisation leads to balanced results. It can lead to improvement compared to the Wasserstein distance, but only after careful tuning of the regularisation parameter; otherwise, it can be very harmful. The best regularisation constant differs for all models. However, for this task, there is usually no validation set to tune this regularisation parameter. Thus, we advise against using this regularisation, or any of the advanced strategies compared in this section. To sum up, the best results with OT are obtained with the Wasserstein distance on top of clustering of the scalable embeddings. The k-means clustering leads to higher performance than affinity propagation on average.

	GEMS	SemEval				DURel	Avg
		English	Latin	German	Swedish		
Averaging	0.349	0.315	**0.496**	0.565	0.212	0.656	0.432
WITHOUT FILTERING AND MERGING OF CLUSTERS							
k-means 5 JSD	0.430	0.316	0.358	0.508	0.073	0.658	0.390
k-means 7 JSD	**0.465**	0.271	0.421	0.498	0.058	0.716	0.405
aff-prop JSD	0.394	0.371	0.346	0.498	0.012	0.512	0.355
k-means 5 WD	0.372	0.360	0.450	0.514	0.316	0.607	0.437
k-means 7 WD	0.375	0.384	0.446	0.503	0.270	0.535	0.419
aff-prop WD	0.369	**0.456**	0.397	0.421	0.264	0.484	0.399
WITH FILTERING AND MERGING OF CLUSTERS							
k-means 5 JSD	0.448	0.318	0.374	0.519	0.073	0.649	0.397
k-means 7 JSD	0.464	0.267	0.432	0.490	0.053	**0.723**	0.405
aff-prop JSD	0.403	0.348	0.408	**0.583**	0.018	0.712	0.412
k-means 5 WD	0.382	0.375	0.466	0.520	0.332	0.628	0.451
k-means 7 WD	0.363	0.375	0.471	0.534	0.307	0.635	0.448
aff-prop WD	0.352	0.437	0.488	0.561	0.321	0.686	**0.474**

Table 3.9 – Spearman Rank Correlation between system output rankings and ground truth rankings for all datasets, with scalable embeddings only. Grey values indicate non-significant correlation (p-value > 0.05).

3.4.3 Clustering, filtering and merging

In this section, we observe the impact of our clusters merging and filtering technique introduced in Section 2.2.4 on the scalable embeddings. After doing the clustering, we select each cluster with less than 10 elements inside, and we merge it with the closest cluster (according to cosine distance between the average of the embeddings inside each cluster). If there is no cluster close enough, the small cluster is considered as an outlier and deleted. The larger clusters are also merged together if their distance is lower than a threshold.

We focus on scalable embeddings, which lead to the best correlation results so far (obtained with WD). We apply our two clustering algorithms, k-means and affinity propagation, and the two distance measures, JSD and WD.

The results can be found in Table 3.9. We add the result of the averaging method for comparison purpose. The filtering method has a large effect on affinity propagation clusters, almost always improving the accuracy. This is explained by the number of clusters and their distribution. As we saw in section 3.4.1, for affinity propagation, the number of clusters can be very high and the distribution of embeddings in these clusters is extremely unbalanced, with a large number of clusters having very few word occurrences. While k-means is the best method without filtering, filtered affinity propagation leads to the best results on average. The effect of the filtering on k-means is positive on average but the difference is thin, as the number of clusters is much lower. Finally,

	GEMS	SemEval				DURel	Avg
		English	Latin	German	Swedish		
LITERATURE							
SGNS + OP + CD	0.347	0.321	0.372	0.712	**0.631**	**0.814**	**0.533**
Nearest Neighbors	0.310	0.150	0.273	0.627	0.404	0.590	0.392
ELMo APD	0.323	**0.605**	-0.113	0.560	0.569	-	0.386*
ELMo averaging	0.323	0.254	0.360	**0.740**	0.252	-	0.386*
ELMo APD+averaging	0.360	0.546	0.036	0.537	0.546	-	0.433*
BERT SCALABLE EMBEDDINGS							
Averaging	0.349	0.315	**0.496**	0.565	0.212	0.656	0.432
APD Canb	0.328	0.483	0.304	0.479	0.226	0.288	0.351
aff-prop JSD + merging	**0.403**	0.348	0.408	0.583	0.018	0.712	0.412
aff-prop WD + merging	0.352	0.437	0.488	0.561	0.321	0.686	0.474

Table 3.10 – Spearman Rank Correlation between system output rankings and ground truth rankings for all datasets. Grey values indicate non-significant correlation (p-value > 0.05). In the LITERATURE section, SGNS+OP+CD and Nearest Neighbors were obtained by running the code from Gonen, Jawahar, Seddah, and Goldberg (2020) on our data. Results on ELMo were taken from Kutuzov (2020).

the filtering step has a larger (positive) impact when using the WD instead of the JSD. On average, the best performing system is affinity propagation with filtered clusters and Wasserstein distance.

3.4.4 Global comparison

In this section, we report the best results from the previous sections and from the literature in Table 3.10. Our methods using scalable embeddings perform better than the standard embeddings; among all aggregation methods, affinity propagation with a merging and filtering step on top of it perform the best. Finally, the best way to measure the distance after the clustering is to use the Wasserstein distance. As a comparison, we report Averaging with cosine distance (CD), APD with Canberra distance (Canb) and JSD in the table.

We add some results from the literature at the top of Table 3.10. SGNS + OP + CD (Schlechtweg et al., 2019) refers to the state-of-the-art semantic change detection method employing non-contextual word embeddings: the Skip-Gram with Negative Sampling (SGNS) model is trained on two periods independently and aligned using Orthogonal Procrustes (OP). CD is used to measure semantic change. The Nearest Neighbors method (Gonen et al., 2020) also uses SGNS embeddings. For each period, a word is represented by its top nearest neighbors (NN) according to CD. Semantic change is measured as the size of the intersection between the NN lists of two periods. Finally, we add the best results from Kutuzov (2020). In his work, he compares ELMo and BERT embeddings with several semantic change detection methods: averaging, APD

with cosine distance, clustering with JSD, and a measure of context diversity. He obtains good but uneven results depending on the corpora using ELMo with APD and averaging. He ensembles these two measures to obtain its best results, by averaging them. In the table, we report his results using ELMo fine-tuned on each corpus, with embeddings extracted from the top layer of the network.

Our methods outperforms the Nearest Neighbors method but is outperformed by a large margin by SGNS+OP+CD which achieves a score of 0.533. We hypothesise that this can be connected to the fact that the sentences in all but one evaluation corpus (COHA) are shuffled. Consequently, BERT models cannot leverage the sequence of 256 tokens as a context, but are limited to the number of tokens in the sentence. The correlation between larger context and better performance of the transformer-based models has been shown on some NLP tasks before (Dai et al., 2019).

We do not have ELMo results on the DURel corpus, thus the average results of the three ELMo rows in the table are not comparable with the other ones. For comparison, our best average results without DURel is 0.432 (affprop WD + merging). The best performing method, SGNS+OP+CD, scores 0.476 without DURel, outperforming both contextualised embeddings methods. Our best method and the best ELMo method perform comparably, but both use very different semantic change measures. Kutuzov (2020) reports that with the same methods (averaging and APD), BERT embeddings lead to worse performance than ELMo, even through ELMo has twice as few parameters. The difference can be related to how the models were trained; Kutuzov (2020) trains ELMo models from scratch on lemmatized Wikipedia corpora, while BERT models are pre-trained on raw text. Yet the SemEval corpora are all lemmatized, leading to a potential advantage for ELMo embeddings. Our experiments with APD on BERT embeddings led to relatively low performances; however, we showed that using APD on scalable embeddings leads to much higher performance compared to standard embeddings. Thus, applying our salable extraction method on top of ELMo could lead to even better results for the APD.

The best average result we report is the one from the last row, with affinity propagation and Wasserstein distance on BERT scalable embeddings. However, it hides a lot of disparities depending on the tests sets. Averaging outperforms clustering for SemEval Latin; APD performs surprisingly well on SemEval English, outperforming everything else; JSD and WD outperform each other alternatively on the different datasets. This disparity does not seem to be linked to language as English (GEMS and SemEval) and German (DURel and SemEval) have different best-performing method. An hypothesis is that it is due to how each method distributes semantic change scores. In Section 3.2, we saw that the different systems can lead to different distributions of semantic change over the vocabulary. For example, the distribution of APD drifts looks like a normal distribution, while the drifts of the averaging method have the shape of an exponential distribution. However, the distri-

bution of semantic change in the gold data might be very different for the various test sets, and also different from the distribution of semantic change in reality. The distribution of gold scores differs a lot: SemEval Latin and German and DURel have uniform gold scores while SemEval English and Swedish and GEMS have skewed distributions towards the left (low scores). Kutuzov (2020) makes similar remarks.

Other results from the literature, relying on word co-occurrences (Gulordava & Baroni, 2011) and topic models (Frermann & Lapata, 2016) on the GEMS dataset are reported in Appendix B. They are both out-performed by our methods.

To evaluate the quality of these results, we can use the value of the inter-annotator agreement as a human-level point of comparison. It is 0.51 on the GEMS dataset, computed using the average of pair-wise Pearson correlations (Gulordava & Baroni, 2011). DURel corpus had an average pairwise correlation of 0.66 between annotators, and the one of SemEval is 0.62 on average. This highlights the difficulty of the task and puts the performance of the best method into perspective.

3.4.5 Qualitative error analysis

We manually check a few examples from the English test sets (SemEval English and GEMS on COHA) to better understand the situations in which the models are mistaken about the true degree of semantic change compared to the gold standard.

3.4.5.1 Examples

We identify several cases of spurious meaning change. One of the tricky cases for our model on the GEMS dataset is the word *neutron*: according to the manual annotation, it is ranked 81st out of 100 and has a stable meaning, while our affinity propagation + JSD system considers it one of the most changed words and ranked it 9th. We apply the affinity propagation algorithm on all embeddings of *neutron* and visualize the biggest clusters using PCA decomposition (Figure 3.5). In two dimensions, there are two clearly distinctive clusters: cluster 36 in the bottom right corner, drawn with pink crosses, which consists only of instances from the 1990s, and cluster 7 drawn with green dots in the top right corner, which consists only of instances from the 1960s. A manual check reveals that the former cluster consists of sentences which mention *neutron stars*. Though neutron stars had already been discovered in the 1960s, they were probably less known and are not represented in the corpus. In any case, a difference in a collocation frequency does not mean a semantic shift, since collocations often have a non-compositional meaning.

The latter distinctive cluster for *neutron*, consisting of word usages from

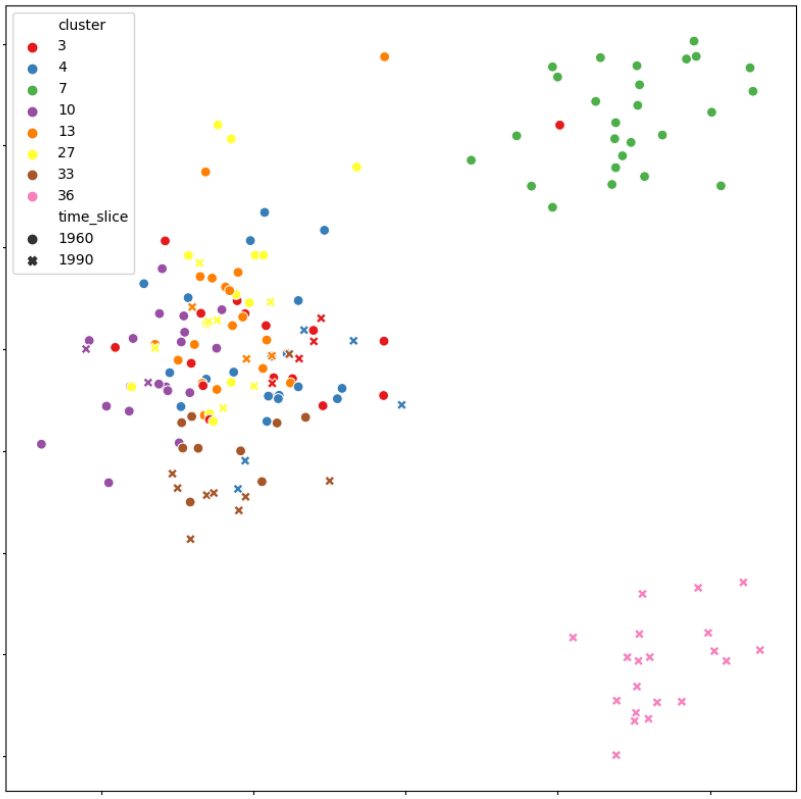


Figure 3.5 – 2D PCA visualization for the biggest clusters obtained for word *neutron* with affinity propagation on standard embeddings from the COHA corpus.

the 1960s, contains many sentences that have a certain dramatic style and elevated emotions, such as underlined in the examples below:

> *throughout the last several decades the dramatic revelation of this new world of matter has been dominated by a most remarkable subatomic particle — the neutron .*
> *the discovery of the neutron by sir james chadwick in 1939 marked a great step forward in understanding the basic nature of matter .*

The lack of such examples in 1990s might have a socio-cultural explanation, or it could be only a corpus artefact. This demonstrates the ability of BERT to capture other aspects of language, including syntax and pragmatics.

Another reason why word embeddings can spuriously undergo semantic change are named entities. An example is the word *vector* in the GEMS dataset. A company called "Vector Security International" appears only in the 1990s time slice, distorting our semantic calculations for this target word. We face the same issue in the English SemEval corpus, with for example the target word *lane*. We see it appearing in sentences such as: *her daddy warn everyone that rose lane be bring home a musician with long hair*. In this sentence, the

target word is used as the surname of a woman, but it can hardly be labelled as a new meaning of *lane*.

Finally, we notice that some clusters contain sentences referring to specific events. For example, one of the clusters for *attack* in the English SemEval corpus contains sentences about terrorist attack in Israel and consists only of sentences from the later time period, for obvious reasons. The sentences in this cluster contain many named entities (NEs), e.g.: *hezbollah leader hassan fadlallah defend* ***attack_nn*** *on israeli civilian target civilian be a war crime*.

All these observations hint at the key role of named entities for this task; detecting them in parallel of performing semantic change detection could improve the results. On top of this, using a pre-trained model with cased vocabulary could prevent many ambiguities for named entities.

3.4.5.2 Spurious semantic change

From observing the results, we identify some possible causes of misleading semantic change compared to the ground truth.

One of the main reasons for a method to detect spurious semantic change is **corpus artefacts**. These are situations where a word suddenly appears in a very specific context in a given period. For example, this can be due to the publication of a book that makes a very special use of the given word. In practice, a corpus can never be perfectly balanced and perfectly representative of language. We can observe the evolution of words in a corpus but extrapolating these observations to the real language has to be done with a lot of precautions.

The examples from the previous sections where target words are used as named entities, such as "Rose Lane" for the target *lane* in the SemEval English dataset, are often cases of corpus artefact.

We can add these limitations to the one observed on the New York Times corpus in Section 3.2; in particular, words which can appear with different part-of-speech seem to have higher semantic change according to BERT models. Similarly, words with very diversified context such as function words can sometimes have spuriously high drift, especially with the APD method.

However, we do not discuss on whether all these cases of doubtful drift should be considered semantic change or not. Our goal is to highlight the limitations of the methods, so that the users can make informed choices about which method to use and what elements should be taken with a grain of salt. In the end, it all depends on the application. For example, a lexicographer might not want to see all the slight context variations in historical corpora appearing as semantic change. On the contrary, a financial analyst observing word usage change in a stream of business news looking for anomalies might be interested.

Post-Processing	English	German
-	0.313	0.436
Target NE	0.328	0.426
Radical NE	0.436	0.302

Table 3.11 – NE Filtering results: Spearman correlation with the ground truth. The method is affinity propagation of embeddings extracted from fine-tuned BERT on two SemEval corpora using the standard method.

3.4.5.3 Named entities cluster filtering

To tackle the issue related to named entities outlined in the examples of the previous sections, we propose two filtering approaches as a post-processing step after the clustering. The first one is the most simple: we identify sentences in which a target word is used as a named entity (NE). We remove a cluster if at least 80% of the target word mentions are NEs. We call it "target NE". In the second one, "radical NE", we filter out clusters where the number of proper nouns is 5 times larger than the number of sentences.

In a real-world application, NE recognition should be done on documents with preserved capitalization, preferably using a model trained specifically on historical documents. Here, we rely on out-of-the-box NLP pipelines.[11] Most of the tools are unable to recognize names in lowercased lemmatized text but POS-taggers are more reliable: e.g., the SpaCy NE recognition model was unable to recognize lower-cased names even if the SpaCy POS-tagger labeled the corresponding tokens as proper nouns.

Results are presented in Table 3.11. The radical NE filtering has a significant impact on English and German results, though in the opposite directions: it improves the performance on the English corpus but reduces it on the German corpus. The effect is the same for the target NE filtering, but to a lower extent. As such, filtering slightly reduces the average performance. The NER systems used for Latin did not perform well and could not recognise entities, while none of the clustering methods led to significant correlation with ground truth for Swedish. That is why we only report results on English and German.

3.5 Binary semantic change task

We open a parenthesis to study the first sub-task of the SemEval 2020 Task 1 challenge: binary semantic change detection. While quantifying the degree of semantic change is a key task that most semantic change detection methods aim at solving, binary semantic change brings complementary infor-

[11] We used SpaCy for English and German (https://spacy.io/), Polyglot for Swedish (https://pypi.org/project/polyglot/) and CLTK for Latin (http://cltk.org/).

mation that may be important for applications such as historical linguistics and lexicography (Schlechtweg et al., 2020). We propose methods to tackle it and present our results.

In order to determine whether a target word has changed or not, we implement two methods. The first one consists in selecting a threshold for the drift measures of all the target words, beyond which we consider that the word is stable or not (*thresholding using stopwords*). The second one uses the clusters to detect a change in the usage of each word, by detecting the appearance or disappearance if a cluster (*identification of period-specific clusters*).

3.5.1 Thresholding using stopwords

We want to find the best threshold in the ranked list of target words by relying on the assumption that stopwords—words that are very frequent in a language and play primarily auxiliary roles—undergo little semantic change.

Though stopwords are more stable than most words of the dictionary, they can still change their meaning due to the grammaticalisation process, i.e. when a previously meaningful word looses most of its functions except the auxiliary ones. For example, the English stopword *hence* used to have a concrete deictic meaning "from here" (e.g. "hence we go") but nowadays it is used only to connect two propositions. Since not all stopwords are stable, finding an appropriate threshold is not straightforward.

It should be noted that stopwords have extremely context-specific representations (Ethayarajh, 2019). However, high polysemy and highly variable context do not necessarily induce more semantic change (Martinc, Montariol, et al., 2020a). We check the difference of average semantic change between a set of stopwords and the list of target words for all languages.

First, to compute semantic change scores for a list of stopwords, we use the same procedure that was used for the target words. For all languages except Latin, we create a list of stopwords by taking the words at the intersection of the `nltk` and `Spacy` stopword lists. For Latin, we use an external resource.[12] We keep only stopwords with more than 30 occurrences in each period; the number of stopwords per language is shown in Table 3.12. When the number of occurrences of a word is too high, we sample 5000 sentences per period for this word. As can be seen in Table 3.12, the mean JSD for the stopwords is sensibly lower than the one for target words.

Then, we compare stopword and target word score distributions in order to define a threshold below which a target word should be classified as unchanged.

We first divide the stopwords' semantic change score distribution into 10 bins to derive a frequency distribution in a histogram with 10 columns, as exemplified for English in Figure 3.6. We take the threshold as the local

[12]List of Latin stopwords: https://github.com/aurelberra/stopwords

		English	Latin	Swedish	German
Number of stopwords		109	334	78	142
Mean JSD	stopwords	0.181	0.210	0.355	0.328
	targets	0.239	0.264	0.460	0.384

Table 3.12 – Number of stopwords used and average semantic change score (JSD) for target words and stopwords.

maximum score of the bin containing a number of stopwords lower than a maximum frequency epsilon ϵ. We exclude the first bin, which is composed of very stable words and can sometimes have a size smaller than ϵ. The maximum frequency ϵ used to select the threshold depends on the number of stopwords for each language: $\epsilon = 1/10 \times$ *number-of-stopwords*. We compute two sets of thresholds: the infimum (lower bound) and the supremum (upper bound) of the bin, as shown in the Figure 3.6. The higher threshold is more conservative, meaning that fewer words will be classified as changed.

3.5.2 Identification of period-specific clusters

The second method looks for concrete indications of semantic change, such as the appearance or disappearance of a specific word sense. All target word occurrences are clustered into a number of distinct clusters; these clusters should to some extent resemble different word senses, allowing identification of target words that obtained or lost a meaning. If one of the clusters for a target word contains word occurrences from one time period and but less or equal than 2 word occurrences from another time period, we assume that this word has lost or gained a specific meaning.

Since clustering methods sometimes produce small-sized clusters, we consider only the clusters bigger than a threshold, in order to focus on the "main" usages of a word. Thus, for k-means we enforce a constraint that a cluster should contain at least 10 word occurrences to be considered in the analysis. For affinity propagation, we implement a dynamic threshold strategy: the threshold beyond which we consider a cluster is computed for each target word as twice its average cluster size.

The idea of determining whether a word meaning has changed or not by identifying time period specific clusters closely resembles the tactic used by the organizers of the SemEval-2020 Task 1 for deriving manual annotations for Subtask 1 (Schlechtweg et al., 2020).

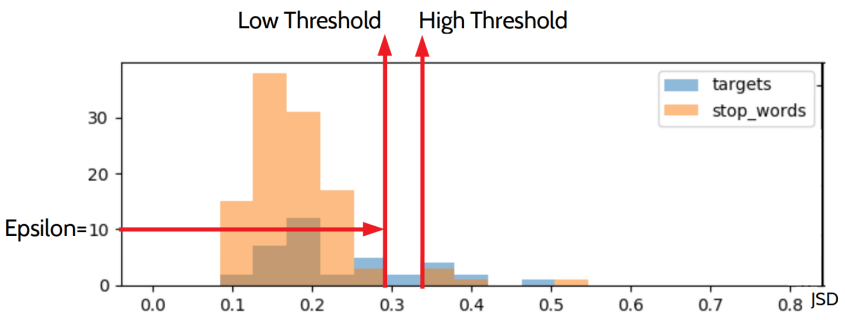


Figure 3.6 – Distribution of semantic change scores in the English corpus: target words VS stopwords.

3.5.3 Results

The results for the binary classification task are shown in Table 3.13. We use only the SemEval corpora since they are the only ones for which we have binary semantic change annotations. We use BERT fine-tuned on the SemEval corpora with the standard extraction method for all results.

The best result was achieved by applying the stopword thresholding method to rankings obtained by ensembling averaging + CD scores and affinity propagation + JSD scores. The ensembling is done by multiplying the semantic change scores produced by the different methods for each target word. We choose multiplication rather than the arithmetic average since the underlying distributions of the semantic shift measures are unknown, even though they produce numbers within the same range. If, for example, the numerical values of a particular measure are generally larger than values of another measure, the former measure would contribute more to the average and thus dominate the ensemble. Multiplication does not have this side effect.

The method of identifying period-specific clusters worked competitively when performing on k-means clusters but performed worse with affinity propagation, since the latter method usually produces a large number of clusters. Reducing the number of clusters by merging the closest clusters together increased the performance of the method.

Looking at the average accuracy, the stopwords-thresholding method performs better than the period-specific clusters method. However, we face high discrepancies between languages. Comparing the results for the same model, i.e. BERT with affinity propagation clustering, the latter method worked best for Latin and worse than the stopwords method for all the other languages.

One would have expected that methods relying on word sense disambiguation, such as our time-specific-clusters methods, would be the most competitive to solve this task. However, among the other systems submitted to SemEval to solve it, the best performing ones all rely on choosing a threshold to discriminate changed words from stable ones, based on a distribution of change scores (same as our stopword-based thresholding method).

Model	Binary method	AVG	English	German	Latin	Swedish
k-means 5	time-specific clusters	0.600	0.649	0.542	0.500	**0.710**
aff-prop	time-specific clusters	0.496	0.568	0.458	**0.700**	0.258
aff-prop + merging	time-specific clusters	0.545	0.514	0.542	0.575	0.548
aff-prop	stopwords, high threshold	0.573	0.622	0.604	0.550	0.516
aff-prop	stopwords, low threshold	0.552	**0.703**	0.667	0.450	0.387
averaging + aff-prop	stopwords, low threshold	**0.621**	0.568	**0.688**	0.550	0.677

Table 3.13 – Accuracy of binary classification on SemEval corpora.

3.6 Use Case: Aylien covid-19 corpus

In the previous sections, we focused on accurately ranking lists of words by semantic change and determining whether a word was undergoing meaning change or not. However, we did not take into account the practical aspect of the task and the potential applications. Concretely, when studying a large corpus in an exploratory fashion, many of the proposed methods are not applicable, requiring either too much memory or too much computational resources (see Section 2.3). Moreover, to understand *how* the meaning of words evolve on top of quantifying it, the sense-differentiating methods—in particular clustering—are more suitable. In Chapter 2, we proposed the scalable extraction algorithm and an interpretation method with the aim of solving these real-life applications.

The combination of scalable clustering with the interpretation pipeline opens new opportunities for diachronic corpus exploration: in the following section, we demonstrate how it could be used to explore the Aylien Coronavirus News Dataset[13]. This corpus contains about 500k news articles related to COVID-19 from January to April 2020[14]. We split it into four monthly chunks and apply the scalable semantic change detection method to explore the temporal word usage differences across months. Table 3.14 shows some statistics of the corpus.

Month	Num. articles	Num. tokens
January	21,102	10,069,362
February	72,057	41,129,710
March	356,983	160,896,860
April	78,690	35,100,352

Table 3.14 – Aylien corpus statistics.

[13] https://blog.aylien.com/free-coronavirus-news-dataset/

[14] We used an old version of the corpus. Currently the data from May is also available.

3.6.1 Identification of the top drifting words

We select a vocabulary of 10k most frequent words. The scalable method allows to perform the token embeddings extraction and k-means clustering for all words of the vocabulary. We extract the top words with the highest average JSD between the successive months to conduct a deeper analysis. We exclude words that appear less than 30 times in each month. It allows us to avoid spurious drifts due to words having too few occurrences in a time slice, such as *kardashian*, and drifts linked to corpus artefacts such as *feb* (diminutive for *February*) which appear every time a date is given, which is quite often in a journalistic corpus. However, some drifts due to corpus artefacts remain, in particular numbers such as *2019-20*. Consequently, we also remove words containing numbers.[15]

Table 3.15 shows the top 10 most drifting words, which were extracted using k-means with $k = 5$ and ranked according to the averaged JSD across the four months. Among them, the words *diamond* and *princess* are related to the cruise ship "Diamond Princess", which suffered from an outbreak of COVID-19 and was quarantined for several weeks. The target word *tiger*, which is the second most changing word, is related first, to the Singaporean flight company Tiger Airways which suffered a lot from the pandemic as it was extensively discussed in February 2020; second, to the Netflix show "Tiger King" which was released in March; and third, to the tigers who contracted COVID-19 in a zoo in the US in April. Thus, the primary context for this word changed several times, which is reflected in our results.

1	diamond	6	bloomberg
2	tiger	7	king
3	costa	8	francis
4	davis	9	ski
5	princess	10	towns

Table 3.15 – Top 10 most drifting words in the Aylien corpus according to monthly-averaged JSD of k-means ($k = 5$) clusters distributions.

3.6.2 Interpretation of the usage change

The interpretation pipeline, described in Chapter 2, is illustrated in Figures 3.7 and 3.8. For example, focusing on the word *diamond* which is the top drifting words in our vocabulary, we cluster its vector representations from the Aylien corpus using affinity propagation and k-means with several values of

[15]This is a rather arbitrary procedure: one can imagine that a domain expert would prefer a different frequency threshold or focus more on a certain month. The most time-consuming part is the embedding extraction. Once this is done, clustering and keyword extraction could be rerun several times in a practical setting according to the needs of the user.

k. Then, using tf-idf, we extract a set of keywords for each cluster from each algorithm and compute the NPMI and TU measures relying on these keywords. Affinity propagation leads to a slightly higher cluster coherence but much lower cluster uniqueness, due to the large amount of clusters, which are often overlapping. Therefore, we use k-means for the interpretation. Lemmatizing the sentences before extracting the keywords leads to more coherent sets of keywords (higher NPMI), and overall better interpretability.

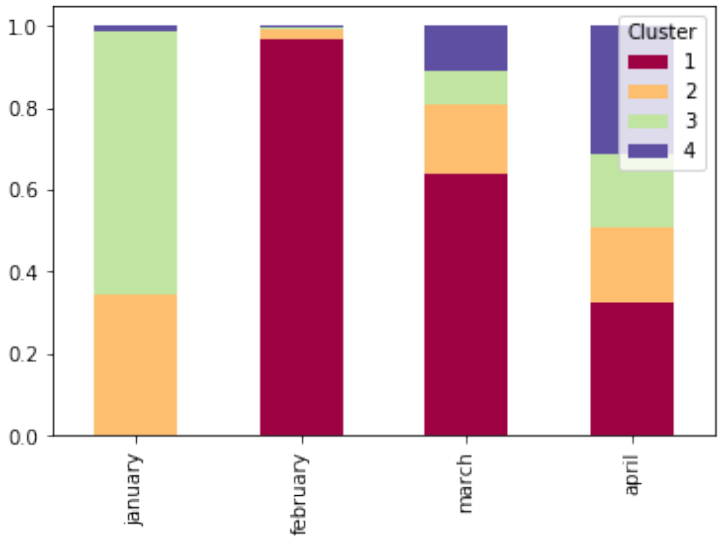


Figure 3.7 – Cluster distributions per month for word *diamond*.

Cluster	Keywords
1	princess, cruise, ship, passenger, quarantine, japan, yokohama, case, board, test, confirm, virus, crew, positive, aboard
2	kong, hong, china, surat, petra, nasdaq, price, sale, low, anglo, face, demand, 000, jewellery, carat
3	south, korea, north, korean, mountain, jubilee, capital, tear, facility, hill, 2012, resort, address, seoul, file
4	sweet, neil, caroline, trump, song, version, house, subscribe, password, nowth, knowthis, jeremy, briefing, play, hand

Slice	JSD
Jan–Feb	0.628
Feb–Mar	0.101
Mar–Apr	0.060

Table 3.16 – Left: cluster keywords for word *diamond*. Right: JSD between consecutive time slices.

A clear temporal tendency is visible from the cluster distribution in Figure 3.7: a new sense appears in February, corresponding to the event of the

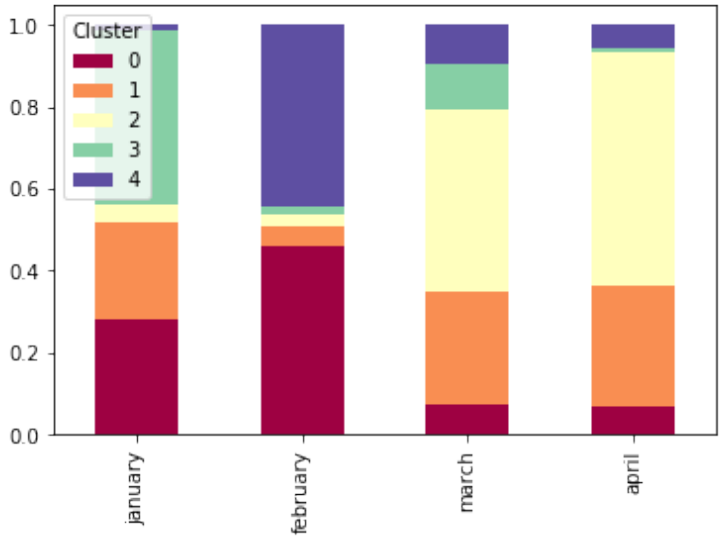


Figure 3.8 – Cluster distributions per month for word *ski*.

quarantined cruise ship (cluster 1); this association is revealed by the keywords extracted from this cluster, listed in Table 3.16. Moreover, the JSD between January and February, when the outbreak happened, is much higher than between the following months when the situation around the cruise ship gradually normalized (Table 3.16, right part). This event, however, had a long echo in the news, as shown by the fact that the cluster 1 is progressively disappearing in March and April. The other clusters deal with other contexts where the word *diamond* appears in the news, in particular related to the struggles of the diamond industry (Petra Diamond company, cluster 2). The cluster 4, that appears in March, is related to Neil Diamond's coronavirus parody of the song "Sweet Caroline" which was shared mid-March on social media platforms and received a lot of attention in the US.

Another example is the word *ski* (Figure 3.8), undergoing important usage change from January to March. This word was used a lot in the news about COVID-19 since ski resorts were major hubs for the virus transmission. The most unevenly distributed clusters correspond to specific events, and their distribution reveals the evolution of their impact on the newspapers over time. The keywords allow us to identify these events, they are listed in Table 3.17. For example, the cluster 0 corresponds to the event of two schools being closed near a French ski resort because of diagnosed cases of coronavirus in a group of UK tourists; it happened at the very beginning of the month of February, and is revealed through the keywords associated with this cluster. Similarly, the cluster 4 corresponds to UK students and staff from several schools being quarantined after returning from ski trips in northern Italy. The cluster 1 is more evenly distributed and is related to the ski season.

Cluster	Keywords
0	resort, french, chalet, singapore, british, virus, briton, stay, case, man, alp, montjoie, contamine, swiss, walsh
1	slope, spa, attraction, entice, resort, economy, boost, restaurant, indoor, idaho, build, shut, school, north, facility
2	resort, area, county, season, loveland, colorado, snow, source, summit, early, mountain, week, vail, 15, virus
3	cup, fis, world, federation, cancel, alpine, race, yanqe, international, yanqing, committee, man, association, 15, olympic
4	trip, school, return, pupil, staff, home, northern, student, flu, newquay, catholic, send, uk, case, fear

Slice	JSD
Jan–Feb	0.282
Feb–Mar	0.312
Mar–Apr	0.038

Table 3.17 – Left: cluster keywords for word *ski*. Right: JSD between consecutive time slices.

3.7 Conclusion

In this chapter, we focused on comparing and evaluating semantic change detection methods. After listing some corpora and evaluation methods from the literature, we conducted two successive analyses, comparing the models introduced in Chapter 2.

The first analysis, which is qualitative, observes the sequential aspects of the models on a corpus of 20 time periods. The second analysis, which is quantitative, relies on annotated data from corpora having only two time periods. These two studies are complementary; annotating a corpus on semantic change sequentially would require a very different method from the ones used for corpora of two periods, and it would be much more complex. The main alternative used by researchers is to generate a synthetic corpus and controlling drifts on a large number of time periods.

From the evaluation on annotated data, we concluded that our best method, which uses the Wasserstein distance on affinity propagation clusters of BERT embeddings, performs worse on average than methods using non-contextual embeddings (Skip-Gram with Orthogonal Procrustes alignment). Despite achieving lower performance, the clustering-based method offers a more fine-grained interpretation than methods based on static embeddings, since it accounts for the fact that words can have multiple meanings. The clustering-based technique returns a degree of change and a set of sentence clusters for each word

in the corpus, roughly corresponding to word senses or particular usages. For this reason, the approach can be used for detecting new word usages, tracing how these usages disappear, and interpreting them; as we have shown with our third analysis on the Aylien corpus of COVID-related news. Word usages and their distributions over time could be linked with real-word events by labeling sentence clusters with a set of cluster-specific keywords. Our analysis on the COVID-19 news corpus allows us to highlight the impact of specific events and their echo across time.

In these experiments, we followed the general approach in semantic shift detection literature and applied our analysis on the raw text. However, our results demonstrate that at least news monitoring applications would benefit from the application of the traditional text processing pipeline, in particular the extraction of multi-word expressions such as named entities and dates. This is confirmed by the error analysis performed after the evaluation on the manually annotated corpora. On top of this, we highlighted some other limitations of the semantic change detection methods relying on BERT embeddings; in particular, the sensitivity to variations of part-of-speech and to intra-period context variability.

Chapter 4

Multilingual analysis

Contents

The goal of this chapter is to extend the analysis of lexical semantic change across two languages. In the previous chapters, we rely on the hypothesis that changes in the context of a word reveal the evolution of the concepts, connotation and concerns associated with this word (Harris, 1954). However, these elements are highly variable depending on the author—his geographical situation, beliefs, the society he belongs to. Thus, the usage of a word can undergo divergences that are meaningful with respect to the divergence between the societies and the people themselves, especially between two languages.

After motivating our work (Section 4.1), we describe the methods we use to learn word representations that are comparable across both time and languages. We use non-contextual word embeddings in Section 4.2 and contextual embeddings from pre-trained multilingual language models in Section 4.3. Then, we suggest a metric to measure divergence of word usage between two languages, that we call *bilingual drift*. We experiment on a synthetic bilingual corpus with controlled semantic change (Section 4.5), before applying our methods to real-word corpora in Section 4.6.3.

4.1 Motivation

All languages do not evolve identically: the impact of an event, the influence of a trend or thinking, can differ between communities. Let us take back two examples from the introduction.

The English word "villain" and is French equivalent "*vilain*" used to refer to village peasants around the 13th century, naturally associated with a lack of good manners and refinement. As time went by, the feudal system disappeared in France and the UK; the original meaning of the word met the same fate. Only the connotation remained, worsening over time towards its current meaning, synonymous with scoundrel and criminal. In this example, the social situation undergoing similar evolution in both countries, the word usage evolved similarly as well.

The second example is the Russian word "*ukrop*." Under the influence of the Russian-Ukrainian conflict, it gained a new meaning in both Russian and Ukrainian languages. However, its usage is pejorative and satyric in the mouth of Russian people, while it is positive in Ukrainian language and later gained a political connotation. This is an example of divergence of word usage in two languages.

This topic is actively researched in the linguistic and sociology research communities (Boberg, 2012), as the scope of language drift offers valuable information for sociological and historical analysis. In the NLP literature, this problem is fairly new. Several authors applied diachronic embeddings models to more than one language (Hamilton et al., 2016; Eger & Mehler, 2016; Rodina et al., 2019). A Semeval shared task took place on this topic, to work on semantic change detection in English, German, Latin, and Swedish (Schlechtweg et al., 2020). However, prior work comparing the evolution of word usage across languages is very limited. Some work study variations between languages or dialects, without looking into the temporal dimension (Hovy & Purschke, 2018; Beinborn & Choenni, 2019). To the best of our knowledge, only Martinc, Novak, and Pollak (2020) conducted bilingual experiments on semantic change, studying the evolution of 4 word pairs in an English-Slovenian corpus of newspaper articles related to immigration: "crime," "economy" "integration" and "politics." They use a BERT model trained on multilingual data, and compute

the cosine similarity of each target word with the word "immigration" over 4 years. The stability of the similarity leads them to conclude that the discourse about immigration is similar in both countries.

More recently, Frossard, Coustaty, Doucet, Jatowt, and Hengchen (2020) propose a resource for analysing the similarities in the evolution of English and French. It is a list of *cognates*—words that have the same etymological origin and similar meaning—in the two languages: 492 pairs of nouns, adjectives and adverbs. They propose a preliminary analysis relying on this resource, focusing on the differences in word frequency over time. They use 1-grams from the Google Books n-grams Corpus, from 1800 to 2008. For each element of the set of cognates pairs, they extract the time series of yearly frequency normalised by the total number of words during the year. They compute the Pearson correlation between the frequency time series of each target word and its cognate counterpart, to compare the patterns of evolution in the intensity of their usage. They also compute the ratio between their maximal and mean frequencies over the full periods, to compare the overall level of word use. They conclude that the cognate words tend to be correlated in terms of frequency over time and have in general a similar level of use in the two languages.

In this chapter, we propose an experimental framework to detect and quantify semantic change in a bilingual setting. We compare *(i) diachronic* and *(ii) contextualised* embeddings, relying on a pre-trained multilingual language model such as m-BERT (Devlin et al., 2019) or XLM-R (Conneau et al., 2020). We also propose an anchored-alignment strategy to tackle the bilingual setting for non-contextual embeddings. Given the absence of dataset annotated with bilingual semantic change, we generate a corpus of synthetic semantic drift across two languages by relying on EuroSense (Delli Bovi, Camacho-Collados, Raganato, & Navigli, 2017), a sense-disambiguated and aligned bilingual corpus. To do so, we define a set of monolingual and bilingual semantic change scenarios and evaluate our different approaches on them. Which allows us to compare the different systems on each scenario, and emphasize the trade-off between their scalability and accuracy.

Finally, we apply our systems to newspaper corpora in two languages, English and French, covering the same time period, from 1987 to 2006. The *New York Times Annotated Corpus* (NYT, Sandhaus, 2008), also used in Chapter 3, collects articles from the New York Times from 1987 to 2006. A corpus of French newspaper articles is constructed ranging over the same time span, from the journal *Le Monde*. We use these corpora to identify words undergoing several kinds of bilingual semantic change that match the bilingual scenarios defined for the synthetic drift generation.

4.2 Diachronic words embeddings

We build two systems based on the continuous bag of words (CBOW) architecture of Word2Vec (Mikolov, Sutskever, et al., 2013) to train bilingual diachronic word embeddings.

4.2.1 Monolingual training

In this section, we consider an hypothetical monolingual corpus divided into T time slices. Each document of the corpus is labeled with its creation date. To train diachronic word embeddings, we rely on a fine-tuning method, as introduced in Section 1.5. We choose not to use an alignment-based method, where a new model would be trained from scratch at each time step (Hamilton et al., 2016). Indeed, when we work on corpora in several languages, an alignment is already needed to map the embedding spaces of the different languages together; it would not be desirable to multiply this type of transformation, as each alignment introduces uncertainty in the system.

To begin with, as advised by Rudolph and Blei (2018), we pre-train our CBOW models on a shuffled version of the full corpus for each language. Then, we build two variants of the system. The first one is *incremental* training (Kim et al., 2014), described in Section 1.5: we incrementally fine-tune the model on each time slice by initialising the embeddings with the ones of the previous time slice. The second variant is *independent* training, where the model is fine-tuned on each time slice independently by initialising it with the pre-trained embeddings. Compared to the incremental method, this one does not take into account the chronology of the corpus and can lead to less directed drifts. However, the fact that the embeddings do not go through a large amount of successive training updates, contrarily to the incremental method, prevents the embeddings from undergoing too extreme drift (Shoemark et al., 2019).

4.2.2 Bilingual alignment

Problem. We now consider an hypothetical bilingual corpus, and embeddings trained separately on each language. We use a bilingual dictionary to create a matching between the vocabularies in the two languages: an injective function mapping words in the first language to words in the second language. However, the vector representations of a given target word and its translation in the other language are not immediately comparable, as they are not in the same vector space.

In order to compare the evolution of a word and its translation jointly, there are two possibilities. The incidental way consists in comparing words inside their vector space: by selecting pairs of words, we can compute their similarity inside each vector space separately. Then, the measures of pairwise

similarity can be compared among languages (Yin et al., 2018). The second approach uses alignments of the two vector spaces, allowing direct comparison of word vectors between the languages. We follow the latter approach and propose the following pipeline for two languages.

Anchoring. The alignment of vector spaces can be unsupervised (for example by automatically inferring a seed bilingual dictionary or using adversarial methods (Conneau, Lample, Ranzato, Denoyer, & Jégou, 2018)) or supervised (using a bilingual dictionary or parallel data). The supervision signal is key to the performance of the overall system, even more than the model architecture itself (Ruder, Vulić, & Søgaard, 2019). In the absence of bilingual dictionary, the signal can come from transparent words in the two vector spaces: words that are the same in both languages, for example named entities. Relying on a bilingual dictionary can raise an issue for our task. Tsakalidis et al. (2019) argue that aligning the vectors of the whole vocabulary is not appropriate for semantic change detection, as it tends to lower the disparities between the different vector spaces. In our case, the alignment forces the embeddings of the word pairs from the supervision dictionary to be the same in the two languages. This might hide some behavior such as a high disparity at the beginning of the full period and a convergence of meanings over time. Consequently, we use a seed dictionary with only the words that we assume are stable during the period. A first set of "stable" words are stopwords (Azarbonyad et al., 2017); however, by definition they do not carry much meaning. Relying only on them for the supervision might result in a poor alignment. We build a complementary set of seed words extracted from a bilingual dictionary, keeping only word pairs that have the same relative frequency in the corpora of each language; with this frequency being in the top 10% of the full corpus, to ensure robust representation. The bilingual dictionary we use for alignment comes from the MUSE tool[1]. It includes 5000 pairs and handles word polysemy.

Alignment. First, we train monolingual CBOW embeddings on each language independently, without dividing the corpora into time slices. To prepare for the alignment, we apply mean-centering to the embeddings of each language, as Schlechtweg et al. (2019) showed the positive impact of this preprocessing step for vector space alignment. For the alignment, we use the system of Conneau et al. (2018) relying on Orthogonal Procrustes (Schönemann, 1966). Orthogonal Procrustes consists in finding an orthogonal map W between two embedding spaces E_1 and E_2, and is presented in more details in Section 1.4. The embedding space of the lower-resource language (source) is mapped to the higher-resource one (target), relying on the bilingual dictionary. The target embeddings stay unchanged, while the map W is applied to the source embeddings, such that all vectors end up in the same space. These aligned embedding vectors are used to initialise the diachronic embeddings,

[1] https://github.com/facebookresearch/MUSE

which can then be trained on all the time slices in both languages, incrementally or independently.

4.3 Contextualised embeddings

On top of CBOW embeddings, we use two pre-trained language models: XLM-R and m-BERT. We summarize how they operate and the key differences between them, before explaining the method to extract multilingual and temporal word representations from them.

4.3.1 Multilingual models

We use two multilingual language models:

- M-BERT, the multilingual version of BERT introduced by Google research;
- XLM-R, a multilingual pre-trained model from Facebook (Conneau et al., 2020).

They are respectively covering 104 and 100 languages, but they also differ on other points.

M-BERT is simply the BERT model trained on multilingual text, without any additional multilingual mechanism nor language identifier. The training corpus is Wikipedia content on 104 languages. Low-resource languages are over-sampled and high-resource languages are under-sampled to deal with the dataset imbalance. The WordPiece vocabulary of 110k tokens is shared across all languages.

XLM-R has the same architecture as RoBERTa (Liu et al., 2019): a Transformer model trained only with the masked language model (MLM) objective. Its input is tokenized with SentencePiece using the unigram language model. It samples streams of text from each language using the same multinomial distribution, to train the model on the MLM task. The main difference with RoBERTa is the scale: it has 550 million parameters—RoBERTa has 355M and m-BERT 110M—and is trained on the publicly available 2.5 TB CommonCrawl Corpus, after a cleaning and filtering step. Moreover, they tune the parameters to generate a huge shared vocabulary of 250k tokens.

XLM-R is claimed to be the first multilingual model to outperform monolingual pre-trained language models. It outperforms monolingual BERT models on many tasks, calling into question the relevance of training monolingual models on low-resource languages. On multilingual tasks such as cross-lingual understanding, it also achieves better performances that m-BERT.

To sum up, two key differences between XLM-R and m-BERT are their size (in number of parameters or in size of the training corpus) and the tokenization system. M-BERT relies on a shared vocabulary of 110k tokens extracted by WordPiece while XLM-R has a 250k tokens vocabulary extracted by SentencePiece.

4.3.2 Extracting word representations

Applying a pre-trained multilingual model on a bilingual temporal corpus allows immediate comparison without requiring any alignment. Each sequence is labelled with the time it was written and its language.

We use the feature-based approach described in Section 2.2, extracting contextualised representations for each token in a sequence. All explanations given in Chapter 2 for the BERT model are identically applicable to the XLM-R model. To sum up all the information about a word from the set of contextual embeddings of all its occurrences in a time slice, we apply two of the aggregation techniques described in Section 2.2:

- **Averaging** of all contextualised embeddings of a word inside each time slice and for each language, leading to a vector $\mathbf{u}_l^{(t)} \in \mathbb{R}^n$ with n the dimension of the embeddings.

- **Clustering** of the contextualised embedding of the word in the full corpus, all language and periods combined. Then, deducing the distribution of clusters at each time slice and for each language $\mathbf{c}_l^{(t)} \in \mathbb{R}^K$ with K the number of clusters.

4.4 Drift measures

After applying all these systems to a bilingual corpus divided into T time slices, for a given target word in a given language l, we obtain several types of information: a sequence of T embeddings $\boldsymbol{u}_l^{(t)}$ in each language (for CBOW and m-BERT / XLM-R with averaging), a vector of sense distributions $\boldsymbol{c}_l^{(t)}$ of length T (for m-BERT and XLM-R with clustering) or T different matrices of token embeddings $E_l^{(t)}$, which are the raw embeddings extracted for each occurrence of a target word (for m-BERT and XLM-R).

We compute the distance between usages as explained in Section 2.4: the APD between matrices of contextualised embeddings, the cosine distance between non-contextual embedding of two time slices, and the JSD between

distributions of clusters.

$$d(t_1, t_2, l_1, l_2) = \begin{cases} \cos(\boldsymbol{u}_{l_1}^{(t_1)}, \boldsymbol{u}_{l_2}^{(t_2)}) & \text{when using averaging} \\ & \quad \text{or non-contextual embeddings} \\ \mathrm{JSD}(\boldsymbol{c}_{l_1}^{(t_1)}, \boldsymbol{c}_{l_2}^{(t_2)}) & \text{when using clustering} \\ \mathrm{APD}(E_{l_1}^{(t_1)}, E_{l_2}^{(t_2)}) & \text{when using no aggregation} \end{cases} \tag{4.1}$$

As to not overload the notations, we remove some obvious parameters such as the word which is being considered, and the method used for embedding extraction. From now on, in order to abstract all these details away, we use the function d as a generic distance between different realisations of an implicit word (in the given languages, at the given time slices).

First, as in the monolingual setting, we use the methods defined in Section 1.6 to measure the drifts of a word in each language: the *incremental* drift, from each time slice to the next one, and the *inceptive* drift, from the beginning of the period to each time slice. We use the full time series to measure the drift instead of only comparing the first and last time slices, as it offers a more fine-grained evaluation. As explained in Section 2.4, we obtain drift vectors in $\mathbb{R}^{T-1}$ for each word in each language, by computing $d(t_1, t_2, l, l)$.

Then, bilingual measures can be computed for each word pair (one word and its translation). First, we compute the distance inside each word pair at each time step. It measures the difference between the usage of a word and its translation at a given time. We call it the bilingual distance $s_B^{(t)} = d(t, t, l_1, l_2)$ for $t = 1, 2, \ldots, T$.

Second, the temporal drift of this distance is measured similarly to the monolingual drift, either *incrementally* or *inceptively*. The distance is the norm between the bilingual distance $s_B^{(t)}$ at two time steps. It allows to detect the convergence or divergence of the usage of a word and its translation. This measure is called the bilingual drift. For example, the *incremental* bilingual drift is computed as follow:

$$D_B^{\text{incr}} = \begin{pmatrix} |s_B^{(0)} - s_B^{(1)}| \\ |s_B^{(1)} - s_B^{(2)}| \\ \vdots \\ |s_B^{(T-1)} - s_B^{(T)}| \end{pmatrix} \tag{4.2}$$

To sum up, the monolingual drifts are series of length $T - 1$ measuring how much the embedding of a word evolves through time in one language. The bilingual drifts are also series of length $T - 1$, measuring how much the distance between a word and its translation evolves through time.

4.5 Synthetic drift generation

We introduced the idea of generating synthetic semantic drift in Section 3.1.3. Using synthetic datasets to tackle the issue of evaluation is common for the semantic change detection task, despite the existence of several annotated corpora. No corpus is available for bilingual analysis. Consequently, we generate a corpus of bilingual synthetic semantic change. It allows us to control exactly the shape and amount of semantic change in the corpus and thus gain a deeper understanding of the impact of each modeling decision. We define a set of bilingual semantic change scenarios and describe the method to build the synthetic corpus.

As explained in Section 3.1.3, we use the natural polysemy of words to generate synthetic drifts as close as possible to reality. To this end, we need a bilingual sense-annotated corpus with consistent annotations between languages (Pasini & Camacho-Collados, 2020). The EuroSense corpus[2] (Delli Bovi et al., 2017) is derived from the Europarl corpus, a large and public corpus of proceedings of the European Parliament. The framework BabelNet (Navigli & Ponzetto, 2012) is used for annotation. The EuroSense corpus has a full and a refined version. We use the latter to build our synthetic corpus; it is half the size of the first one but more reliable. EuroSense contains parallel text in 21 European languages. We focus on the two languages with the highest amount of annotations in the refined corpus: English and French. An example of aligned sentences in these languages can be found in Table 4.1.

	English	French
Sentence	The best tools for this are liberalisation and freer competition , which causes train companies to take a greater interest in the wishes of customers .	*Les meilleurs moyens d'y parvenir sont la libéralisation et une concurrence plus libre , qui incite les compagnies ferroviaires à se soucier davantage des souhaits de leurs clients .*
Lemma	customer	*client*
Sense	bn:00019763n	bn:00019763n

Table 4.1 – Example of aligned sentences in English and French in the EuroSense corpus, with annotated anchor and corresponding sense in the BabelNet framework.

4.5.1 Semantic change scenarios

In order to generate and capture variations of distributions of word senses through time and across two languages, we define several scenarios of word

[2] http://lcl.uniroma1.it/eurosense/

usage variations. Then, we simulate each of them using sentences extracted from the EuroSense corpus. We apply the different systems described in the previous section to evaluate and compare their ability to capture bilingual semantic change.

First, we choose two monolingual scenarios of semantic change (labeled "M") and generate them using sentences extracted from the EuroSense corpus. Assuming we have a target word with at least two senses, the scenarios are:

- $M0$: all senses are fully stable.
- $M1$: one sense gradually appears / disappears, the others stay stable.

Then, we define several scenarios of semantic drift across two languages (bilingual scenarios, labeled "B") derived from the monolingual scenarios. Assuming we have a target words w_1 and its translation w_2 with at least two senses in common:

- $B0$: w_1 and w_2 are $M0$ (stable).
- $B1$: w_1 is M0, w_2 is $M1$ (one stable, the other drifts).
- $B2$: w_1 and w_2 are the same $M1$ (they gain/lose the same sense).
- $B3$: w_1 and w_2 are different $M1$ (one gains/loses one sense, the other gains/loses another sense, while the other common senses are stable: the two words diverge).

4.5.2 Building the synthetic corpus

Step 1: selection of target lemma pairs.

For all the sense-annotated lemmas in English and French in EuroSense, we extract their sets of senses. We only keep the senses with enough occurrences (more than 200 occurrences per language). We associate English and French lemmas together if they have at least two senses in common, creating a bilingual dictionary. From these lemma pairs, we extract the set of sentences annotated with one of the senses in common to build the pool of sentences for the next step.

Even though the sentences are parallel, this step is not straightforward: in many cases, the annotated word is not the same in the two languages. Sometimes, the annotated word is the same but the sense is different. These situations are due to the annotations in EuroSense being automatic. Some cleaning is necessary; to avoid having to discard all the sentences in this situation, we identify the word and its translation in the other language and we assume that if they appear in parallel sentences, then they carry the same sense.

At the end of this step, we have a sense-annotated parallel corpus with all the sentences containing a sense-annotated target lemma. We divide the

target lemma pairs into low polysemy lemmas (number of senses = 2) and high polysemy lemmas (more than 2 senses). This distinction allows us to evaluate the various systems according to the degree of polysemy of the target lemmas. For example, a low-polysemy lemma pair (English, French) is (*project*, *projet*) and a high-polysemy one is (*measure*, *mesure*).

In total, we have 115 English-French lemma pairs, of which 66 have 2 senses and 49 have between 3 and 5 senses.

Step 2: creation of sense distributions.

For each monolingual scenario, we create probability distributions of senses at each time slice. Let's denote by $p(S \mid \mathcal{T}, W, L)$ the probability that the lemma W conveys sense S at time $\mathcal{T}$ in language L. We choose to generate $T = 10$ time slices and apply each scenario to all the target lemmas pairs. Since our variables are discrete, for a given lemma w in language l, the probability distribution of a set of 2 senses $\{s_1, s_2\}$ over time can be characterised by a $2 \times T$ left stochastic matrix[3]:

$$\begin{pmatrix} p(s_1 \mid \mathcal{T} = 1, w, l) & p(s_1 \mid \mathcal{T} = 2, w, l) & \cdots & p(s_1 \mid \mathcal{T} = T, w, l) \\ p(s_2 \mid \mathcal{T} = 1, w, l) & p(s_2 \mid \mathcal{T} = 2, w, l) & \cdots & p(s_2 \mid \mathcal{T} = T, w, l) \end{pmatrix}.$$

More generally, when a word carries $m \geq 2$ senses, their distribution can be specified by a $m \times T$ matrix.

To be more precise, for a given target lemma: for the $M0$ scenario, we randomly draw an initial distribution over the set of senses and repeat it at each time slice: $p(S \mid \mathcal{T} = t, w, l) = p(S \mid \mathcal{T} = 1, w, l)$ for $t = 2, 3, \ldots, T$. For the $M1$ scenario, we randomly draw a starting time, a shape of evolution (linear or logarithmic) and a trend (increasing or decreasing). We select one of the senses of the lemma and gradually increase or decreases its probability of appearance through time after the starting point. The other senses have stable distribution across time.

Step 3: creation of the synthetic corpus.

For each monolingual scenario, we build the synthetic corpus time slice after time slice, using the set of target lemmas, the pool of sense-annotated sentences and the distributions of senses.

For each target lemma, at each time step, we sample 200 sentences for each of its senses. Then, we add each sampled sentence to the corpus for the current time step, with the probability specified in the corresponding distribution of senses of the scenario. To avoid the synthetic sense distribution for a target lemma to be disturbed by noise from its appearance as a context word in other sentences, when adding a sentence to the synthetic corpus, we attach the suffix "_l" to its target lemma.

All the bilingual scenarios are built from the monolingual ones. Generating

[3]Meaning that its columns sum to 1.

them simply reduces to using the right monolingual scenarios for each word and its translation. For example in the $B3$ scenario, we generate a corpus using the $M1$ scenario for both the target lemma and its translation, but select a different sense to appear or disappear in order to induce a divergence.

The obtained corpora, for each scenario and each language, have around 7.5M words distributed into the 10 time slices.

4.5.3 Evaluation method

At each time t, a word w in a language l is characterised by its sense distribution in the synthetic corpus $p(S \mid t, w, l)$. This information is similar to the cluster distributions extracted when applying clustering to contextualised embeddings; we can compute the drift measure defined in Section 4.4, using the JSD to compare the sense distributions. The drifts obtained from these measures can then be used as gold standard for the evaluation of our systems.

For each system described in the previous sections and for each target lemma pair, we output the monolingual drifts computed on the monolingual scenario synthetic corpora and the bilingual drifts computed for the bilingual scenarios (see Section 4.4). We wish to evaluate whether these series have the same trend as the gold standard.

For this, we use the Mann-Kendall (MK) Trend Test (Kendall, 1975; Sen, 1968). It is a non-parametric statistical test used to detect trends of variables. It is particularly suited to monotonic trends, which is how we designed the semantic drift in our data. The null hypothesis of the test is the absence of monotonic trend. If the detected trend in our data is the same as the one from the gold standard drift (no monotonic trend, upward trend, or downward trend), we consider the semantic change for the target lemma has been correctly identified. We then compute the accuracy (both for monolingual drift and bilingual drift) as the proportion of correctly identified trends in the full list of target lemmas.

4.6 Experimental results

We apply our systems to the synthetic corpus and to real data (newspaper articles) to evaluate them and analyse their behaviour.

4.6.1 Experimental setup

As we rely on stopwords (on top of frequent words) for the alignment, we do not discard them during preprocessing. We count the word frequencies in both languages to build the dictionary for the alignment, relying on the

external bilingual dictionary.

CBOW processing: The context size is set to 5 words, and the dimension of word embeddings to 50. Preliminary experiments with larger embeddings dimension exhibited no significant improvement. We posit this is due to the small size of the dataset, so we decided to keep the dimension of the embeddings low. We train all models using 10 epochs.

For each language, a static model is first trained on the set of all sentences containing the target lemmas. Then, we proceed with the training of the different systems.

BERT and XLM-R processing: We use the pre-trained *bert-base-multilingual-uncased* and *XLM-roberta-base* models from the `transformers` library. We extract the contextualised embeddings from the corpus and apply the different aggregation methods. We remove the "_l" suffix of the target lemmas before extracting their embeddings. Note that the clustering step has to be done on both languages jointly, as the distribution of clusters needs to be comparable between the two languages.

4.6.2 Results on synthetic data

Model	Diachrony	Stable $M0$	Drift $M1$	Both stable $B0$	Stable&drift $B1$	Same drift $B2$	Diverge $B3$
CBOW	incremental	0.65 - 0.16	0.54 - 0.96	0.87 - 0.82	0.66 - 0.46	0.76 - 0.68	0.63 - 0.47
	independent	0.84 - 0.83	0.63 - 0.86	0.83 - 0.89	0.70 - 0.45	0.80 - 0.66	0.67 - 0.50
BERT	averaging	0.86 - 0.87	0.34 - 0.55	0.84 - 0.90	0.79 - 0.4	0.71 - 0.69	0.63 - 0.47
	k-means 5	0.85 - 0.86	0.61 - 0.19	0.86 - 0.97	0.78 - 0.41	0.77 - 0.91	0.66 - 0.40
	APD	0.88 - 0.87	0.54 - 0.26	0.82 - 0.89	0.78 - 0.33	0.65 - 0.73	0.67 - 0.44
XLM-R	averaging	0.86 - 0.85	0.35 - 0.56	0.84 - 0.96	0.76 - 0.43	0.68 - 0.76	0.74 - 0.54
	k-means 5	0.85 - 0.89	0.61 -0.18	0.86 - 0.96	0.74 - 0.38	0.76 - 0.89	0.75 - 0.54
	APD	0.77 - 0.87	0.52 - 0.26	0.80 - 0.93	0.74 - 0.34	0.67 - 0.81	0.70 - 0.53

Table 4.2 – Description (embedding type, diachrony solution) and accuracy measure of each system compared with gold standard in each semantic change scenarios. The numbers of the left are *incremental drift* while the ones on the right are *inceptive drift*.

Table 4.2 summarises the accuracies measured using the Mann-Kendall trend test (Hussain & Mahmud, 2019), as described in section 4.5.3. It compares the drift of all systems with the gold standard drift, for each monolingual and bilingual scenario. We choose $k = 5$ clusters for k-means, as it is the maximum number of senses that can be found in our list of target lemmas. Experiments with higher values of k did not improve the accuracy.

We have three scenarios with stable monolingual or bilingual drift ($M0$ and $B0$, with all the words being stable; and $B2$, with words in both languages drifting in the same direction) and three drifting scenarios ($M1$ and $B1$, where one sense drifts; and $B3$, where words in both languages drift in different

directions). The results show that stable scenarios are generally easier to detect accurately compared to the changing ones, especially in the monolingual analysis.

The best results are obtained with BERT using k-means clustering. All the information is summarised into a vector of size K (here 5), representing the different uses of the word; it might provide a better focus on the meaningful changes in word usage. XLM-R leads to slightly lower results for most scenarios. In the case of CBOW, independent training lead to better performances than incremental training. This is in line with the findings of Shoemark et al. (2019): the large amount of training updates, especially in such a small corpus, is harmful for the quality of the representation.

Overall, the *inceptive* drift measure lead to better accuracy for stable scenarios, while the *incremental* drift is more suited to scenarios where the sense distributions evolve or diverge.

Finally, we compare the ability of the various systems to handle highly polysemous words. We divide the set of target lemma pairs into low (2 senses) and high-polysemy lemmas (more than 2 senses) and compute the correlation of the drift from each system for the two groups of lemmas, with the gold standard. The average accuracies across all scenarios are in table 4.3, respectively labeled "LP" and "HP." The accuracy on high-polysemy lemmas is significantly higher than low-polysemy ones for the methods relying on pre-trained language models. Methods relying on CBOW embeddings have a much smaller disparity between low and high-polysemy words, even out-performing the other methods except for the BERT k-means model.

Model	Diachrony	HP	LP
CBOW	incremental	0.64 - 0.61	0.53 - 0.58
	independent	0.65 - 0.62	0.60 - 0.60
BERT	averaging	0.57 - 0.53	0.52 - 0.53
	k-means 5	0.71 - 0.52	0.60 - 0.47
	APD	0.66 - 0.50	0.55 - 0.47
XLM-R	averaging	0.65 - 0.56	0.53 - 0.56
	k-means 5	0.68 - 0.51	0.58 - 0.46
	APD	0.65 - 0.51	0.54 - 0.47

Table 4.3 – Average accuracy for low and high-polysemy words (LP and HP). The numbers on the left are *incremental drift*, the ones on the right are *inceptive drift*.

4.6.3 Real data description and setup

We analyse the drift in a bilingual corpus of news articles, using the CBOW embeddings with incremental training. Indeed, even though using BERT with token embeddings clustering lead to better results overall, the extraction of

each token embeddings and the clustering step are computationally heavy: it is more suited for a fine-grained analysis of the senses of a limited set of target words rather than a global analysis on the full vocabulary.

The *New York Times Annotated Corpus* (Sandhaus, 2008), introduced in more details in Section 3.1.1, gathers around 1 855 000 articles from January 1987 to June 2007. We scrape *Le Monde*, one of the most read daily newspapers in France, on the same time period. We divide both corpora into $T = 20$ yearly time steps and select a vocabulary containing the $V = 40\,000$ most frequent words for each corpora. The average number of words is around 3.5 M for one time step in the French corpus and 9 M in the English one. The experimental setup is the same as the one used on the synthetic corpus; the volume of data being higher, we increase the capacity of our model by setting the dimension of CBOW embeddings to 100.

4.6.4 Detection of bilingual drift

First, a bilingual lexicon is built using the intersection of the MUSE bilingual dictionary with the French and English vocabularies from our corpora. We manually update the bilingual lexicon with domain-specific vocabulary such as named entities, in order to improve the coverage on the corpora. The final bilingual dictionary has 27 351 words.

We pre-train the CBOW model on both corpora and normalise the embeddings to prepare for the alignment. The French corpus being the smallest, its embeddings are mapped to the English embedding space. Then, we incrementally update the aligned embeddings on both corpora. For each word of the bilingual vocabulary, we compute its monolingual inceptive drift and its bilingual drift as defined in section 4.4. It allows us to identify the words belonging to each of the bilingual scenarios that we defined, using the average drift in each language and the average bilingual drift as thresholds. The proportion of each scenario as well as some example words are in Table 4.4. The words that are stable in both languages ($B0$) are mostly daily life words (e.g. *mayonnaise*). The words that drift in the same direction in both languages ($B2$) are concepts related to technology and society that are common to the English and French culture (e.g. *renewable*); while the words that diverge between the two languages ($B1$-fr (English stable, French drifting), $B1$-en and $B3$) belong to more culture-specific concepts (e.g. *francs*) or controversial topics (e.g. *terrorist*). For example, *francs* drifts in French, while it is stable in English. This is probably due to the large change of currency in France in 2002 that had much lower media coverage in the US. Similarly, *terrorist* drifts in both languages but in different directions. The two countries went through many terrorist attacks during the period under study, but from very different groups, leading to different contexts for this word.

We applied the systems using BERT with the different aggregation methods to the 20 word pairs with the most extreme measures for each scenario

(highest values of monolingual drift and bilingual drift, except for scenario $B0$: lowest values). The classification of the word pairs into bilingual drift scenario is sensibly the same between the methods relying on BERT and the results described in Table 4.4, except for scenario $B2$ (words that drift in the same direction on both languages). Instead of $B2$, the BERT systems classify these words as $B1$ or $B3$, disagreeing with CBOW embeddings on the stability of the bilingual distance.

$B0$	$B1$-fr	$B1$-en	$B2$	$B3$
58.2%	15.5%	16.2%	4.9%	5.2%
dinosaurs	reforms	bush	genomics	steroid
pottery	delinquency	horrific	renewable	rockets
anniversaries	francs	maid	condom	gay
mayonnaise	feminine	hostages	cinemas	katrina
joke	provincial	dealers	robotic	terrorist

Table 4.4 – Proportion and example words for the different categories of bilingual drift.

4.7 Discussion

In this chapter, we studied the joint evolution of words in two corpora of different languages, by defining an experimental framework to characterise the drift of the distance of a word and its translation, relying on several bilingual drift scenarios. We compared several systems using static and contextualised embeddings.

The use of BERT contextualised embeddings, coupled with a clustering step to aggregate the information from all the contextualised embeddings of a word, leads to the best performance. The performance of the CBOW embeddings, especially with independent training, is nevertheless very competitive. Non-contextual embeddings are often shown to perform well in diachronic tasks (see Section 3.4).

Overall, there is still a large margin for improvement; be it in terms of quality of bilingual representation, metric to measure bilingual semantic change, or evaluation measures. In particular, the underlying bilingual representation learning approach is key for the detection of drifts. The transformations and methods used to create a cross-lingual word embedding space might result in information loss or generation of spurious drifts in the embeddings. Future work could focus on finding methods to better represent bilingual embeddings with the purpose of detecting bilingual semantic change. The anchored alignment method presented here is not the only option; Temporal Referencing (Schlechtweg et al., 2019), the Global Anchor method (Yin et al., 2018), or any other method for alignment, could be good candidates.

A limitation of our work is that it does not take into account the interactions between languages, which are an important component on bilingual semantic change. Modeling this interaction could be beneficial for this task. Another limitation is the use of an injection to define word pairs. In his general linguistics course, De Saussure (1916) states there is no bijective relationship between words in different languages. First, the different meanings and uses of a word in a language cannot have a perfectly identical equivalent in another language. And second, as noted by Frossard et al. (2020), a word can have one or more synonyms in one language while the word bearing the same meaning in another language has none; in that case, the usage of the word in the first language is divided into all its synonyms.

Finally, a crucial aspect of this task is evaluation. We use synthetic data to generate semantic drifts in the two languages. Synthetic evaluation is common in monolingual semantic change analysis, even though there is no guarantee that the generated phenomenon is similar to real-world data. It has several limitations; in our case, as we build all bilingual scenarios from combinations of two monolingual scenarios, the flaws of the monolingual scenarios are inherited by the bilingual scenarios. It can potentially multiply the noise by propagation of uncertainty. We wished to overcome the limitations of synthetic evaluation with the application on real corpora, but more thorough interpretation would be necessary for a qualitative evaluation. To perform quantitative evaluation, an annotated dataset similar to the ones for monolingual semantic change of Section 3.4 would be necessary. However, the annotation task would be even more complex than for monolingual data. An easier entrance point towards annotating data for this task could be borrowed words (words adopted by the speakers of one language from a different language). Overall, this is a challenging task and we hope to attract more people to work on it in the future.

Chapter 5

Word usage variation in the financial domain

Contents

This PhD was funded by a bank, Société Générale, in the context of a CIFRE contract. As is the case in all other areas, the banking domain faces domain-specific data associated with domain-specific tasks.

In this chapter, we underline the specificity of financial data, with a focus on annual financial reports of companies. We provide examples of associated tasks, along with a literature review on the topic of semantic change for financial data (Section 5.1).

A key NLP task is the search for *early warning signals*: pieces of information extracted from text, that may have ambiguous implication but provide

valuable information for the understanding of present events and the prediction of future ones, if interpreted correctly. Financial analysts look for this kind of indicators by monitoring newspapers and statements or reports from financial actors. In Section 5.2, we propose to use semantic information through word usage change as a signal. We apply our semantic change detection methods to financial data—two corpora of annual activity reports from companies, and a corpus on central banks statements—to study the kind of information that can be extracted and interpreted.

In particular, we extend our semantic change detection pipeline to other dimensions than time. We saw in the previous chapters that the methods relying on contextualised embeddings do not use time as an ordinal variable: each time slice can be considered as a category independent from the other time slices. Consequently, we can apply our methods to any dimension on top of time: for example, in a corpus of company reports, we observe how words vary depending on the activity sector of the company. Each sector is considered as a category, exactly as we deal with time slices. It allows us to observe how word usage vary across different financial actors, business lines, or any other dimension; and to interpret this evolution. Parts of this section were made in collaboration with Pr. Asanobu Kitamoto from the National Institute of Informatics (NII, Tokyo).

Finally, in Section 5.3, we observe the link between semantic change information and classical financial indicators used by domain experts. We focus on companies stock prices, and compare it with time series of semantic drift of company names.

5.1 NLP in the financial domain

Traditionally, financial experts and economists have used quantitative information for financial analysis and decision making. However, in the financial domain, textual data is plentiful and carry a lot of potentially useful information. This major resource is usually analysed manually by experts. However, the volume of textual data increased tremendously in the past decades with the progressive dematerialisation and the growing capacity to share and store data (Lewis & Young, 2019). This data was often overlooked and left without large-scale analysis. Recently, the "FinTech" industry started to thrive. Its goal is to improve financial activity with technology, and it often involves Natural Language Processing methods.

5.1.1 Financial textual data

Textual data in the financial domain are from diversified sources and can take different shapes. A large amount of textual data comes from financial newspapers. A lot of information can also be extracted from social media

posts made by financial actors such as corporate insiders, investors, customers, suppliers, pressure groups and politicians. The financial data released by companies gather all elements of their periodic reporting package, such as their periodic activity reports, their financial statements, and their regulatory filings. It also include their press releases and social media posts. Another valuable source of information are speeches, statements and reports by policymakers, regulators and politicians. All the aforementioned documents are public data; however, companies and organisations also have many internal textual documents such as reports from financial analysts.

Among all these data sources, periodic corporate reporting receive a particularly large attention from the research community, with an already plentiful literature in the financial domain and a growing one in NLP. The reports have to be made publicly available periodically by all companies above a certain size and market value threshold, defined by regulatory authorities of each country. Their content is also controlled by the regulators.

One of most widely studied type of company report are 10-K filings (Dyer, Lang, & Stice-Lawrence, 2017). They are annual reports required by the U.S. Securities and Exchange Commission (SEC). They include information about the activity of the company throughout the year and a description of its financial performance, as well as more generic information such as the structure and the history of the firm. Quarterly reports (10-Q) as well as 8-K reports, only published for reporting events which might be of importance to shareholders—such as a change in the board members or bankruptcy—are also required by regulators. All SEC filings can be found online on the EDGAR database.[1]

10-K filings are so diligently studied thanks to their format, which is highly standardised and controlled by the SEC. Outside the US, companies periodic reporting is less standardized and more shareholder-oriented. It is the case in France, where companies only publish their reports on their websites, with no aggregation by a regulatory authority. Nevertheless, public corpora of French companies annual report were constructed by researchers: the CoFiF Corpus (Daudert & Ahmadi, 2019) includes financial reports of the 60 largest French companies from 1995 to 2018, while the DoRe Corpus (Masson & Paroubek, 2020) contains reports from 336 French and Belgian companies from 2009 to 2019. In this chapter, we conduct analyses on the CoFiF corpus as well as on a corpus of 10-K filings. We perform analyses in the DoRe corpus in Appendix D.

Lewis and Young (2019) report significant increase in the size and complexity of UK annual report narratives: their median number of words more than doubled between 2003 and 2016[2] while the median number of items in the table of contents also doubled in the same period. Similar behavior is observed in 10-K filings in the US (Dyer et al., 2017). This phenomenon observed in finan-

[1] https://www.sec.gov/edgar/searchedgar/companysearch.html

[2] For a sample of 19,426 PDF annual reports published by 3252 firms listed on the London Stock Exchange.

cial reporting from companies leads to a contradiction: the huge increase in volume leads to more and more need of solutions from the NLP community for analysing this unstructured data automatically. However, more reporting from more companies leads to more diversity in the shape of the documents; this lack of standardization and structure makes the analysis tougher and requires more complex methods (Lewis & Young, 2019).

5.1.2 Financial NLP tasks

Computational methods for financial text processing allow analysts to process large amounts of data, and more specifically to extract information and detect patterns and latent features that would be tricky to identify otherwise. With the growing interest of financial companies and organisations towards such computational methods for finance, several workshops were organised in the recent years: among others, we can cite Economics and Natural Language Processing (ECONLP) and Financial Narrative Processing (FNP) in 2018, and Financial Technology and Natural Language Processing (FinNLP) in 2019.

In association with these workshops, several shared task were proposed to encourage researchers to work on these topics. These tasks showcase problems which are crucial for the financial domain but remain tricky to tackle. For example, the following shared tasks were all organised in the year 2020 alone: FinSBD-2 (Sentence Boundary Detection in PDF Noisy Text in the Financial Domain), FinSim (Learning Semantic Representations for the Financial Domain), FNS (Financial Narrative Summarisation), and FinTOC (Financial Document Structure Extraction). We can cite other common financial tasks:

- Stock price prediction and market trend analysis, using newspapers or tweets (low-granularity data), often associated with sentence classification and sentiment analysis.
- Tasks associated to risk management (e.g. credit risk, operational risk). Common tasks for this field are document classification, relation extraction and anomaly detection.
- Asset management, Know Your Customer (KYC), compliance, fraud detection...

In the domain of fraud detection, we conducted some experiments on the French corpus of financial reports CoFiF (Daudert & Ahmadi, 2019) mentioned in the previous section. We proposed an extension of the sense clustering and interpretation method from Chapter 2 to detect omissions of risk factors in companies annual reports. This work can be found in appendix D and in our paper (Masson & Montariol, 2020).

For all these tasks, one must take into account the specificity of financial data and of the financial domain. Financial data are characterized by domain-specific terminology: words that have a fixed meaning in general settings but

may appear in a different context in a specific domain. For example, the word "*liability*" has a negative connotation in general; however, it is a common financial term that does not carry any connotation in a financial text. To deal with this issue, strategies range from domain-specific vocabulary lists to transfer learning. A common step is to fine-tune language models on domain-specific data. For example, Desola, Hanna, and Nonis (2019) fine-tuned BERT separately on two corpora of SEC-EDGAR filings (from years 1998–1999 and years 2017–2019) to create the model FinBERT.

As in many other domain-specific application such as justice or health, most analyses in the financial domain face the crucial need of transparency, explainability and fairness. A common example is machine learning-driven credit risk assessment in banks: the goal is to use automatic methods to help analysts deciding whether to approve or deny a loan. It is crucial to ensure the fairness of the algorithm and the explainability of loan approval or refusal, as required by regulators.

5.1.3 Semantic change on financial data

From a diachronic point of view, we are only aware of Purver, Valentincic, Pahor, and Pollak (2018), who study 20 years of financial reports of 30 Dow Jones Industrial Average (DJIA) companies, from 1996 to 2015. They manually select a set of 12 financial terms and investigate changes in lexical associations, by looking at the evolution of the similarity between pairs of two terms. More recently, Desola et al. (2019) studies two corpora of SEC-EDGAR filings (from years 1998–1999 and years 2017–2019). For three selected words (*cloud*, *taxes* and *rates*), they compare the embeddings from the two periods using cosine similarity. Neither of these works are fully unsupervised nor allow a fine-grained analysis of the word usage variations.

A less sense-oriented application is the work of Buechel, Junker, Schlaak, Michelsen, and Hahn (2019): they study a two-decades corpus of public statements from two central banks, the European Central Bank (ECB) and the US Federal Reserve Bank (Fed). They use emotion analysis to show how the emotions identified in the statements evolve with the economic situation across time. They consider the emotional dimensions of Valence, Arousal and Dominance (VAD model). Figure 5.1 shows the dominance series for both central banks in the full period, allowing to compare the behaviour of the two actors and juxtapose it with key events such as change in presidency or economic crisis.

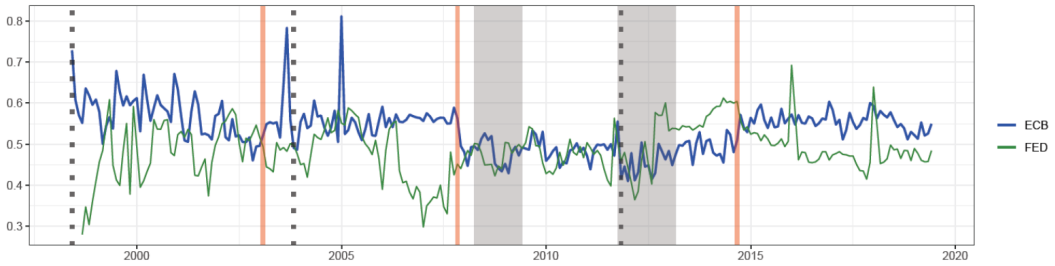


Figure 5.1 – Dominance series for ECB (blue line) and Fed (green) (Buechel, Junker, Schlaak, Michelsen, & Hahn, 2019). Vertical dotted lines indicate beginning of ECB presidency, shaded areas indicate Euro area recession periods.

5.2 Detecting variation in word usage in financial data

In this section, we leverage the capacity of the semantic change detection methods based on contextualised embeddings to analyse financial texts along different axes of variation. Indeed, financial texts are characterised by many domain-specific terms and entities whose usage is subject to high variations, reflecting the disparity and evolution of the opinion and situation of financial actors. Starting from a corpus of central bank statements and two corpora of annual company reports spanning 20 years, we explore the ability of our semantic change detection methods to identify and interpret variations in word usage in the financial domain.

5.2.1 Motivation

As stated in the introduction, variation in word usage is not limited to temporal evolution. The usage of a word can vary depending on the person that uses it: several dimensions (geographical, cultural) can lead communities to use words in a different way depending on the local interests and concerns. These two kinds of variations can be denoted as *diachronic* (through time) and *synchronic* (across any other dimension than time: the country, the source, the author...).

In the literature, the analysis of synchronic variation is mostly done through domain-specific word sense disambiguation (WSD). Some approaches use similarity measures between non-contextual word embeddings to analyse the variations in word usage among several communities (Tredici & Fernández, 2017; Ferrari et al., 2017). More recently, Schlechtweg et al. (2019) analyses both diachronic and synchronic drifts using non-contextual word embeddings with vector space alignment. Using one corpus spanning different periods (texts from the 16th to the 20th century) and one corpus spanning different domains (cooking-related texts from several categories), they demonstrate that the di-

achronic models and evaluation can be successfully applied to the synchronic detection or word usage variation.

In the financial domain, detecting the variations in word usage through time can lead to better understanding of the stakes and concerns of different periods (Purver et al., 2018). In a synchronic way, many dimensions can be observed: how the words are used depending on the business line, the country of origin, the company or organisation that produces the document... This way, the opinions, behaviours and preoccupations of the writer can transpire through its specific usage of words.

This semantic variation information can be useful to financial analysts. It can help to better understand the variations of concerns and viewpoints of financial actors, for example by analysing text from regulatory authorities. It can also help to identify the impact of an event on different actors through time, using high temporal granularity data sources, such as business news articles or tweets. Additionally, it can provide information to analyse the evolution of a crisis, by tracking the evolution of the usages of financial keywords and detecting new drifting words.

In other words, we look for *weak signals* through the scope of word usage change. A weak signal is an element observed from data that has ambiguous interpretation and implication, but may be of importance in the understanding and prediction of events (present or future). In the financial domain, any change in strategy, emerging concern or unusual event linked to a financial actor can be a weak signal; identifying relevant weak signals and interpreting them is an extremely challenging task.

In this section, we study word usage change as a potential signal of evolution in the situation and opinion of a financial actor. When an analyst reads a set of financial documents, the diachronic and synchronic variations in word usage are not immediately visible. But they might reveal valuable information, if they can be detected and interpreted. For example, it can be shown that the connotation of the vocabulary used by central banks in their reports and statements is strongly influenced by the economic situation (Buechel et al., 2019), despite the fact that they attempt to be as neutral as possible due to their position as regulatory authorities; to avoid worrying the investors, among other reasons.

We use the detection and interpretation pipeline described in chapter 2 to determine in a fine-grained way the different kinds of use of a word and the distribution of these uses in a financial corpus. Our goal is to analyse financial texts in a diachronic and synchronic way, as a preliminary investigation to address the following questions:

1. In a synchronic way, what do word usages reveal about the opinion and behaviour of different financial actors?

2. In a diachronic way, what does it says about their evolution? Can it

improve our understanding of past and ongoing events through the scope of word usage change?

3. Is word usage change related to financial indicators? How complementary are these information? (This last item is tackled in Section 5.3).

5.2.2 Pipeline

For all the words of the vocabulary that undergo word usage variation, we want to answer the following questions:
• For each dimension under study, how much does the usage of the word vary?
• At what time does a usage drift happen (for the diachronic dimension) or which actor has a significantly different usage distribution?
• What is the change about?

We rely on the pipeline described in Section 2.2, using clustering of contextualised embeddings to distribute all the occurrences of a word into clusters of usage. We use the clustering to interpret the different usages of a word and how they vary across a dimension, using centroid sentences and detecting keywords for each cluster (see the interpretation methods in Section 2.4).

However, this clustering method can only be performed on a limited set of target words due to scalability issues (complexity and memory consumption, see details in Section 2.3.1). Thus, a preliminary step selects the set of target words—words that have highly variable usage—before the clustering and interpretation steps.

1. Preliminary step: target words selection. For all words in the vocabulary, we compute a variation metric for each of its dimensions of variation. It is an extension of the *averaging by time slice* metric D_w from Section 2.3.2 to any dimension: we take the average pairwise distances between the average embeddings of each class of the dimension under study. For example, if we wish to study the influence of the source of the documents for a given target word, we compute the average word embedding for each different source. We compute the average pairwise distance between these average embeddings, and select only the subset of the vocabulary with the highest distance—the highest variation of context in the dimension—for the clustering step.

2. Clustering. We apply two clustering methods, k-means and affinity propagation. Each occurrence of a target word is labelled according to the dimension under study (the time slice where the token appears, the category of the document...). We construct the probability distributions over the clusters of usages of a target word for each class of the dimension, as exposed in section 2.2 for the diachronic case.

3. Quantification and interpretation. We use the Jensen-Shannon divergence (JSD) and its generalisation to n probability distributions, applicable in both synchronic and diachronic cases. We compare each distribution with the average distribution of the full dimension to detect the classes with the highest variations.

In order to capture the clusters involved in the variation, we identify the ones that have an uneven distribution across all the elements of the dimension. It allows us to extract information such as which cluster is specific to a given actor or which clusters vary the most. Finally, we can get an interpretation of the usages associated with the clusters using the methods from section 2.4.

5.2.3 Experiment on central banks statements

Corpus description. This corpus assembles all the official statements of two central banks, the European Central Bank (ECB) and the US Federal Reserve Bank (Fed) from June 1998 to June 2019.[3] These statements report the economic situation and expose the policy decisions of the central banks. This corpus was Web-scraped and studied through emotion analysis by Buechel et al. (2019). It is composed of 230 documents from the ECB and 181 from the Fed, and contain a total of 14 604 sentences; it is heavily unbalanced towards the ECB (more than 75% of sentences), as the Fed statements are usually shorter. An example of the style of these documents can be found in Figure 5.2.

> *Compared with the March 2019 ECB staff macroeconomic projections, the outlook for real GDP growth has been revised up by 0.1 percentage points for 2019 and has been revised down by 0.2 percentage points for 2020 and by 0.1 percentage points for 2021. The risks surrounding the euro area growth outlook remain tilted to the downside, on account of the prolonged presence of uncertainties, related to geopolitical factors, the rising threat of protectionism and vulnerabilities in emerging markets.*

Figure 5.2 – Extract of ECB statement from June 6, 2019.

Selection of target words. We divide the corpus into yearly time steps, and build the vocabulary from all words having at least 100 occurrences, excluding stopwords.

We conduct a preliminary step on the full vocabulary to identify the most changing words. To speed up the process, we sample at most 3 000 sentences for each word. We extract the embeddings of the target words using the English pre-trained bert-base-uncased model from the `transformers` library. Then, we

[3]We thank Sven Buechel from Jena University Language & Information Engineering (JULIE) Lab for sharing the corpus with us.

	Time	Source
1	households	measures
2	labor	committee
3	holdings	rate
4	securities	employment
5	accomodative	developments
6	sectors	support
7	monetary	pressures
8	housing	price
9	sales	stability
10	loan	market

Table 5.1 – Top 10 words with highest average pairwise variation measure for the time dimension and the source dimension on the Central Bank Statements corpus.

compute the average pairwise variation measure by year and by source. The words with highest variation for the two dimensions are listed in Table 5.1. For the source dimension, we keep only the words with a threshold of presence of at least 50 occurrences per source. The comparison is hampered by the fact that some words such as *labor* are absent from the FED statements because of orthographic divergence between British English and American English; in that case, we can only study their variation across the temporal dimension. For each dimension, we select the 10% words with highest variation measure as target words for the clustering step.

Comparison of the clustering algorithms. We apply both k-means and affinity propagation on the set of contextualised embeddings of each target word. In the case of k-means, we try different values of the number of clusters k ranging in $[\![2, 10]\!]$. To evaluate the quality of a clustering, we compute its silhouette score. Then, we extract the distributions of the clusters across each dimension (for example the distribution of the clusters inside each time slice for the time dimension). Finally, we apply the generalised Jensen-Shannon Divergence (JSD) on the set of probability distributions to measure the level of usage variation of the word. The average values of silhouette score, JSD by source and JSD by year for all target words of this corpus for different algorithms are summarised in Table 5.2.

As a reminder, the silhouette score measures the density of the clusters and the distance between them. It falls between -1 and 1 and must be maximised, with a score of 0 indicating overlapping clusters. Additionally, while the JSD between two distributions takes values between 0 and 1, the generalised version to n distributions is bounded by $\log_2(n)$. For the temporal dimension in the Central Bank Statements corpus, the 20-years period leads to an upper bound being equal to $\log_2(20) \approx 4.32$.

Method	S-score	JSD-synchronic	JSD-diachronic
Aff-prop	0.267	0.829	2.519
KMeans 3	0.213	0.342	0.523
KMeans 5	0.215	0.467	0.856
KMeans 7	0.218	0.537	1.088

Table 5.2 – Clustering quality (Silhouette score) and average JSD on the Central Bank Statements corpus for source and time dimension, for all target words. We use affinity propagation and k-means with different k.

According to Table 5.2, the average silhouette score is the highest for the affinity propagation algorithm. The average JSD for both dimensions increases with the number of clusters for the k-means algorithm, as a higher number of clusters naturally leads to larger entropy values. We also inspect the number of clusters for the affinity propagation algorithm: it ranges from 4 to 450, with an average number of 61 clusters.

The word *households* in the Central Bank Statements corpus has the highest temporal variation according to the preliminary measures (average pairwise variation, Table 5.1) and indeed, we find that it also has a high JSD on the temporal dimension. We compute affinity propagation clustering, as it leads to the highest silhouette score. However, the number of clusters outputted by this algorithm is too high to allow for a clear interpretation of the clusters. Thus, we apply a merging strategy similar to the one described in section 2.2.4. Using a threshold of 10 embeddings inside clusters, we merge the minority usages (*illegitimate* clusters) with the closest big ones until no small cluster is left. We end up with 13 large clusters of usage. We plot the distribution of the merged clusters on Figure 5.3. The normalisation of word count inside the clusters allows an easier comparison of the proportions but hides the disparities in frequency; that is why, in this figure, we show the un-normalised counts to also highlight the evolution of the global trend of the target word.

The cluster number 13 is mostly appearing in the years 2009 and 2010. Its centroid sentence is *"we expect price stability to be maintained over the medium term, thereby supporting the purchasing power of euro area households."* The other sentences of the cluster also have positive connotation. This cluster is composed in majority by sentences from the ECB statements. The phenomenon might be linked with a need from the European Central Bank to be reassuring during that period of recession after the financial crisis of 2009 that struck the European countries.

On the other hand, cluster number 12 is mostly present from 2017. Its centroid sentence is *"the annual growth rate of loans to non-financial corporations increased to 2.9% in October 2017, after 2.4% in September, while the annual growth rate of loans to households remained stable at 2.7%."* Other sentences of the cluster also talk about loans; This cluster emerges after 2017 following new guidelines from the European Central Bank in this domain.

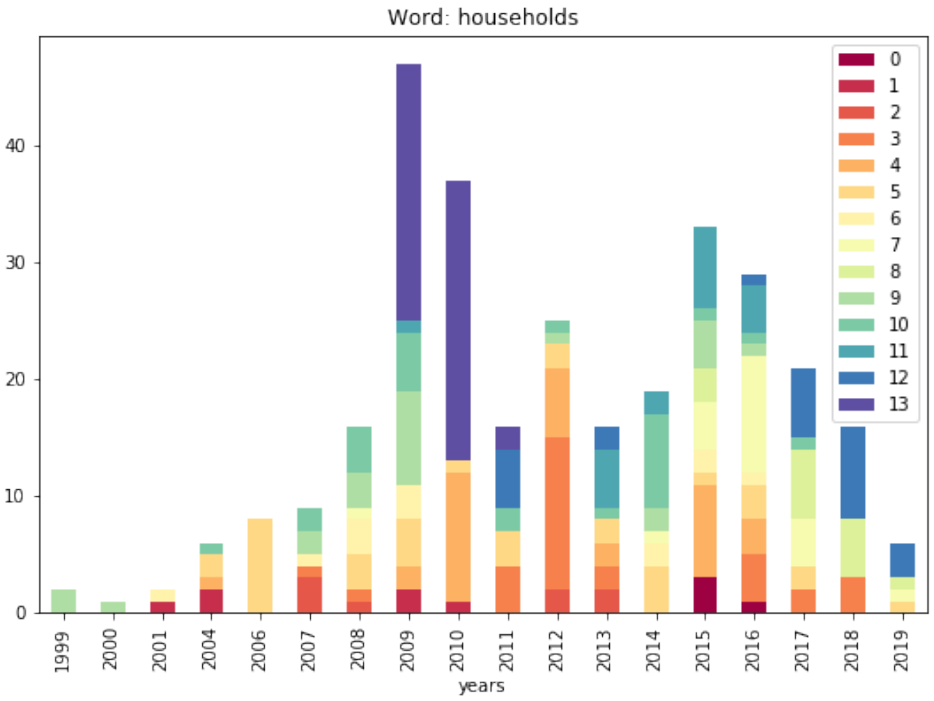


Figure 5.3 – Distribution of clusters per year for the word "households" in the Central Bank Statements corpus.

5.2.4 Experiments on annual reports

We continue our experiments with two corpora of companies annual reports, in English and French. Compared to the central banks corpus, they have more diversity in the synchronic dimension, covering a large number of companies from diverse business lines and stock exchanges. It allows us to dive deeper into synchronic analysis; we apply the word usage variation detection pipeline across several dimensions in addition to time. For French data, we use FlauBERT (Le et al., 2020), a variation of the language models BERT and RoBERTa (Liu et al., 2019). Its pre-trained version FlauBERT-base-uncased is trained on French data from a wide variety of sources, from encyclopedic data to newspaper articles.

5.2.4.1 Data: US and France annual reports

The English corpus comprises annual financial reports (10-K) of US companies extracted from the Securities Exchange Commission Edgar database.[4] We collect[5] the 10-K reports from the 500 biggest companies in the US, between 1998 and 2018. Similarly to Purver et al. (2018), we extract the Part I and the Items 7 and 7A from the Part II of the reports. These sections mainly

[4] https://www.sec.gov/edgar.shtml
[5] Using https://github.com/alions7000/SEC-EDGAR-text

describe the activity of the company and its operations and management. We end up with 8 676 documents spanning two decades, amounting to a total of 7.3 million sentences. Figure 5.4 illustrates the style of these documents.

> *On average, our high inventory velocity means we generally collect from consumers before our payments to suppliers come due. We expect variability in inventory turnover over time since it is affected by numerous factors, including our product mix, the mix of sales by us and by third-party sellers, our continuing focus on in-stock inventory availability and selection of product offerings, our investment in new geographies and product lines, and the extent to which we choose to utilize third-party fulfillment providers.*

Figure 5.4 – Extract of Amazon 10-K filing Item 7 ("Management's Discussion and Analysis of Financial Condition and Results of Operations") from January 30, 2020.

Each document is written by one company, and for each company, we extract additional data: its stock exchange (NYSE, NASDAQ, OTC) and its Standard Industrial Classification[6] (SIC) code. The latter indicates the business line of the company; the classification is divided into 7 Offices and sub-divided into 444 industries. All the offices are listed in Table 5.3. Thus, we can detect drifts across several dimensions, from the most to the least fine-grained: by company, by industry, by office, and by stock exchange.

Label	Description	%
0	Office of Energy & Transportation	15.1
1	Office of Finance	12.5
2	Office of Life Sciences	14.7
3	Office of Manufacturing	19.7
4	Office of Real Estate & Construction	8.2
5	Office of Technology	13.1
6	Office of Trade & Services	16.7

Table 5.3 – Label and proportion of business line with SIC classification in the SEC-Edgar corpus

The French corpus is taken from the CoFiF financial corpus[7] (Daudert & Ahmadi, 2019). It is composed of the financial reports of the 60 largest French companies belonging to the CAC40 and CAC Next 20 stock market indices and contains more than 5 million sentences in 2 655 reports of different types (quarterly, half-yearly, annual and reference document reports), from 1995 to

[6]Described in https://www.sec.gov/info/edgar/siccodes.htm
[7]https://github.com/CoFiF/Corpus

2018. The year is not included in the corpus metadata, and is not consistently mentioned in the title of each document. We therefore automatically extract all the years mentioned in the title and in the first lines of each document, and deduce the year of publication of the report from the intersection of these two lists. We exclude the years 1995 to 1998 for the analysis, because of too low volume. A particularity of this corpus is the presence of raw data tables in the text. In order to exclude these elements from the analysis, when dividing the corpus into sentences, we exclude sentences composed of less than 70% letters (more than 30% numbers, symbols and spaces). Finally, we focus on reference documents (RDs) only, which make up almost 85% of the data volume. They are published annually by companies and summarize their financial situation and prospects.

We end up with a corpus of about 2.7 million sentences. Each of these sentences is associated with the metadata of the document from which it is extracted: the name of the company, and the year of publication of the report. The synchronic axis is expanded by collecting information about the companies: their sector (luxury, transport, chemicals...) and their domain of activity (secondary or tertiary).

5.2.4.2 Results on usage distribution and interpretation

We conduct a preliminary step on the full vocabulary to identify the most changing words, as we did for the central banks corpus. In both corpora, some words appear with very high frequency; for example, the word *million* appears 1.4 million times in the SEC-Edgar corpus. To speed up the extraction process, we sample 3000 sentences for each word. We extract the embeddings of the target words using BERT or FlauBERT. Then, we compute the average pairwise variation measure for all dimensions under study: by year, by industry, by office, and by stock exchange for the SEC-Edgar corpus, and by year, field of activity and sector in the CoFiF corpus. We do not compute the variation by company, as the granularity is too high; the volume of data (only one report per year) leads to a term frequency too low to distinguish drifting words from non-drifting words.

We perform the same analysis as for the central bank statements corpus, selecting target words, extracting the probability distributions of clusters in a dimension, and computing the generalized JSD to measure the semantic variation. For both corpora, we present examples for the diachronic dimension and for the synchronic dimension in order to show the different possibilities in terms of interpretation.

Example on the CoFiF corpus: the word *ecology*. The affinity propagation algorithm leads to the highest average silhouette score; we use it to rank the variations of the target words. The 5 target words with the highest JSD along the time and sector dimensions in the CoFiF corpus are listed in Table

5.4. Note that the 3 most variable words per year are all part of the climate lexical field. Moreover, the word *ecology* varies strongly in both dimensions; we further analyze and interpret its variation.

	By year		By sector	
1	écologie	*ecology*	magasin	*shop*
2	climat	*climate*	écologie	*ecology*
3	biodiversité	*biodiversity*	luxe	*luxury*
4	syndicats	*union*	syndicats	*union*
5	gouvernement	*government*	publicité	*advertising*

Table 5.4 – Top 5 words with largest JSD for the time and sector dimensions, with affinity propagation clustering, in the CoFiF corpus.

For this word, the highest silhouette score is obtained from the k-means algorithm with $k = 7$. The normalized distributions of clusters resulting from this clustering for the sector and time dimensions are shown in Figure 5.5. By comparing the distributions of the clusters in each period or sector with the mean distribution on the corpus for the dimension, we identify the periods and sectors that stand out. Then, we quantify the variation of each cluster within a dimension. This leads us to the last step: cluster interpretation. We extract the central sentence of each cluster (the one closest to the centroid). Finally, using the keyword extraction method described in Section 2.4, we associate a theme to each cluster in Table 5.5.

For example, Cluster 6 has a strong temporal variation, with an increasing proportion since 2007; it is associated with financing and cost issues (Figure 5.5). Cluster 2 is specific to only a few sectors and focuses on business ideas specific to ecology; it appears quite late in the period. Conversely, Clusters 1 and 5, respectively specific transportation and land development for one, and energy for the other, are common to most sectors. Cluster 1 is well summarized by its central sentence, "these obligations are mainly related to ecology, landscape planning, and archaeology for the associated development sites." Cluster 3, associated with the concept of territory from the resources point of view, is present across the full period but is specific to only a few sectors such as oil and chemistry. Finally, Cluster 4 contains the lexical field of risk and danger. It appears once at the beginning of the period and is more and more frequent at the end of the period. The appearance of this cluster in 2001 is probably due to noise, as the 2001 slice has only two occurrences of the target word. This cluster remains in the minority even at the end of the period, probably due to the fact that financial analysts avoid using negative terms when writing financial reports so as not to worry investors.

Examples on the SEC-EDGAR corpus. For the synchronic dimension, we study the distribution of usages of the word *client* by office (business line). It is one of the words with the highest JSD for this dimension. The silhouette

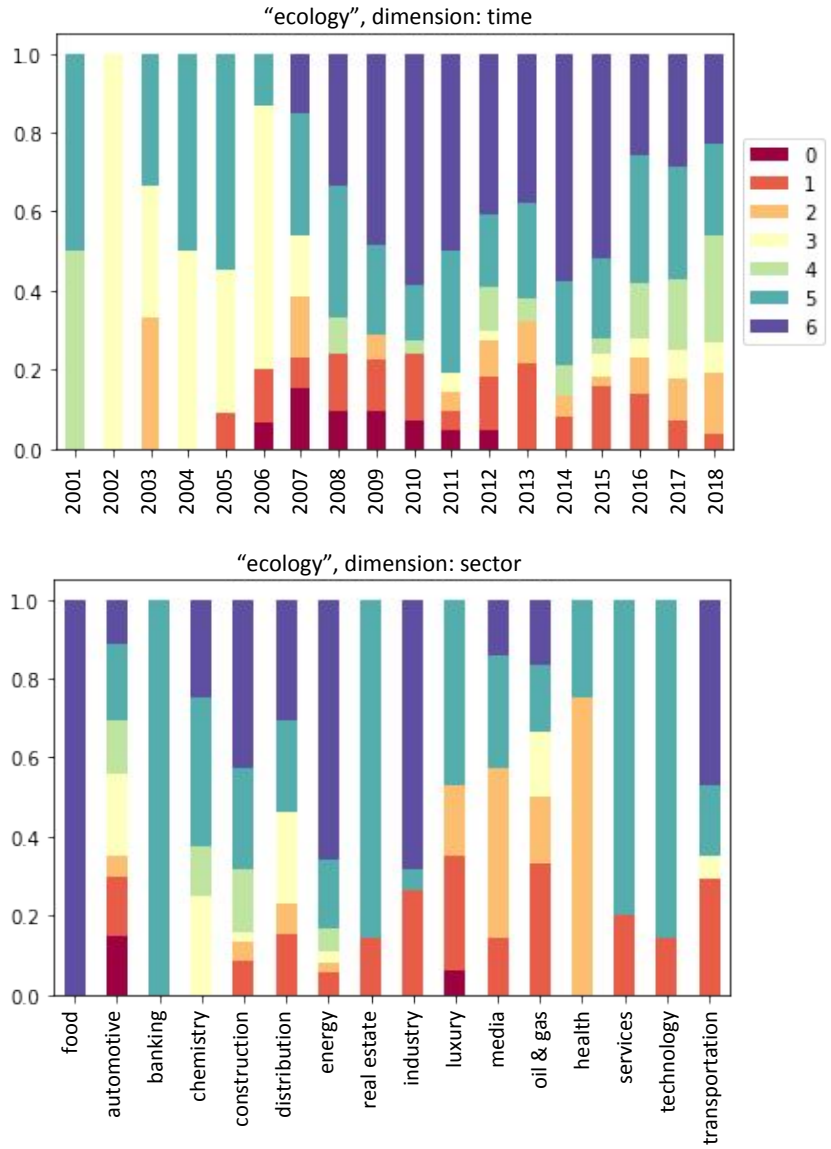


Figure 5.5 – Distribution of clusters for word *ecology*, by sector (bottom) and year (top).

score is the highest using k-means algorithm with $k = 4$. All the offices are listed in Table 5.3; The normalised distributions of clusters for each of them are in the upper part of Figure 5.6. Using the keyword extraction method, we select the most representative words for each cluster an report them in Table 5.6. Cluster 1 is the most unevenly distributed, and appears mostly in documents belonging to the Real Estate & Construction Office. The keywords associated with this cluster involve the idea of paying (*cost*, *fees*) and negativity (*risk*, *loss*). On the contrary, Clusters 2 and 3 are relatively similarly allocated in the different offices. Their keywords correspond to very classical vocabulary from financial reports. Finally, Cluster 0 is characterised by vocabulary from the semantic field of digital technologies (*server*, *applications*...): the clustering algorithm was able to identify this specific meaning of the target word.

Title	Example of keywords
0-practical	éco, concept, logement, économique, raisonné, préservant, préfabrication *eco, concept, housing, economical, reasoned, preserving, prefabrication*
1-transportation	directeur, énergie, impacts, transports, aviation, initiatives, territoire, aménagement *director, energy, impacts, transportation, aviation, initiatives, territory, planning*
2-job	apprendre, structure, métiers, collaborateurs, réseau, professionnels, management *learning, structure, jobs, collaborators, network, professionals, management*
3-territory	industrielle, sites, flux, déchets, échanges, territoriale, eaux, circulaire, ressources *industrial, sites, flows, waste, exchanges, territorial, water, circular, resources*
4-danger	groupe, fondation, prix, intégrer, péril, polluante, excessive, concernés *group, foundation, price, integrate, peril, polluting, excessive, concerned*
5-energy	émissions, énergie, fessenheim, industrielle, biodiversité, slovénie, co2, nucléaire *emissions, energy, fessenheim, industrial, biodiversity, slovenia, co2, nuclear*
6-cost	énergie, arrêté, coût, mer, prix, stockage, économiques, milliards, aménagement *energy, stopped, cost, sea, price, storage, economic, billions, development*

Table 5.5 – List of clusters and interpretation for word *ecology* in the French CoFiF corpus.

For the diachronic dimension, we study the distribution of usages of the word *crisis* by year (Figure 5.6, bottom). The highest silhouette score corresponds to the k-means algorithm with $k = 5$. The keywords for these 5 clusters can be found in Table 5.7. We can identify clear temporal tendencies in the figure. The proportions of Clusters 0 and 4 are decreasing through time, while Clusters 1 and 2 are growing. The extraction of keywords allows to differentiate the 5 usages of the word *crisis*. For example, Cluster 1 is associated with vocabulary of the domain of marketing and media. It is almost non-existent before the year 2004, and is rapidly growing. Cluster 2 is related to the crisis of the debt of the European countries; it appears and grows after 2008. Cluster 3 can be found across all the period; it is associated with slightly negative words (*accident* and *loss*), similarly to Cluster 4 (associated with *debt* and *recession*) whose proportion decreases after 2010.

However, one has to be wary of the selection of the number of clusters using the silhouette score. Sometimes, it leads to choose a low amount of clusters that may hide some valuable information. For example, for the target word *insurance*, the silhouette score is maximum for k-means with $k = 2$. However, using $k > 5$, a cluster appears that belong mostly to sector 4 (Office of Real Estate & Construction); it is associated with the keywords *property* and *investment*, showing a new aspect of the concept of insurance specific to this sector.

5.2.5 Discussion

We showed that using contextualised embeddings associated with clustering allows to automatically detect variations in the use of a word across several dimension.

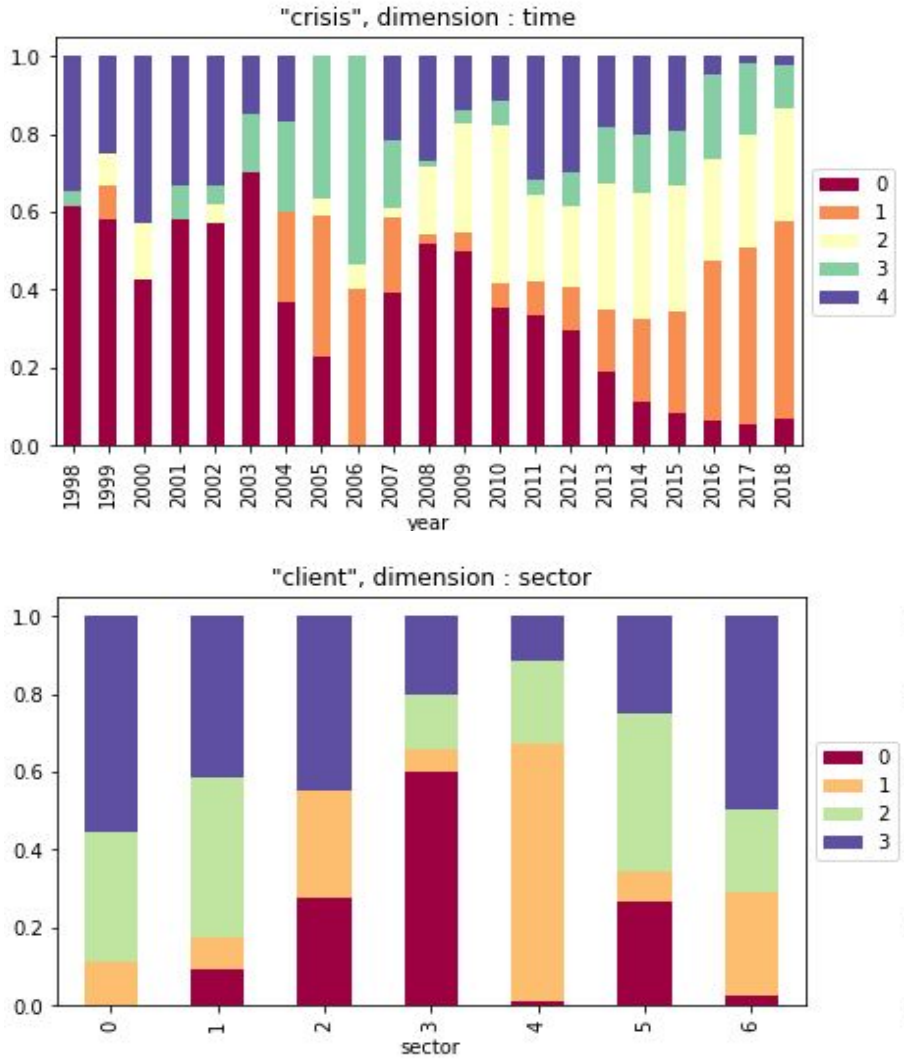


Figure 5.6 – Distribution of clusters per Office for the word *client* (bottom) and per year for the word *crisis* (top) in the SEC-Edgar corpus. The Offices are described in Table 5.3.

Overall, the distinctiveness and consistency of vocabulary and connotation inside each cluster is encouraging. The clustering allows us to identify variations in meaning as well as usage. In particular, the ability to detect clear temporal tendencies in the cluster distributions could allow a financial analyst to link these clusters with real-world events, and have a deeper understanding of the phenomena behind them. The presence of cluster that seem to be characterised by a common sentiment opens the track for further sentiment-specific analysis.

However, even though the keyword extraction method allows us to gain insight on the interpretation of the clusters, it still requires some domain-specific knowledge. The next step is to link the detected word usage variations with

Cluster	Keyword examples for the word *client*
0	server, products, data, applications, services, systems
1	revenue, contract, risk, costs, loss, business, fees
2	assets, funds, cash, interest, balances, investment
3	services, business, revenue, growth, management, products

Table 5.6 – List of clusters and keyword examples for the word *client* in the SEC-Edgar Corpus.

Cluster	Keyword examples for the word *crisis*
0	liquidity, funding, contingency, cash, collateral, outflows
1	marketing, business, management, design, advertising, media
2	european, debt, credit, sovereign, countries, eurozone, banks
3	financial, accident, capital, regulatory, loss, liquidity, funding
4	credit, financial, global, markets, debt, european, recession

Table 5.7 – List of clusters and keyword examples for the word *crisis* in the SEC-Edgar Corpus.

numerical indicators. First, it could offer a better understanding of the implications of the variations of word usage and complement their interpretations. Second, it would help to build an evaluation framework for our methods. For example, we can observe jointly the cluster distributions of the token embeddings of the word *unemployment* by office in the SEC-Edgar reports, and the actual unemployment time series by office.

We propose in the next section a preliminary analysis to compare word usage change and financial indicators. Instead of company annual reports, we use daily newspaper articles which offer longer time series, allowing a more robust analysis. We compare embeddings of company names in the newspapers with the stock prices of these companies.

5.3 Linking semantic change with numerical time series

The goal of this section is to quantify and analyse the relationship between semantic information from text and financial indicators. We use the same corpus as in Section 3.6: COVID19-related news articles. The health crisis has a huge financial and economical impact on society, in particular on industry. This is visible on the evolution of stock market values of publicly traded companies. We link it with the usage evolution of the name of these companies.

5.3.1 Experimental framework

Data. We use the last version of the Aylien corpus[8] introduced in Section 3.6. It contains 1.6M of Coronavirus-related news articles from November 2019 to July 2020, in English, from 440 sources which are mostly business newspapers. The amount of news for the first two months being low, we consider only news from January to July. We divide the corpus into 1-day time slices. Named entity recognition has been performed to detect entities in each sentence. Focusing on companies, we select the 100 most frequent companies in the corpus traded on a stock market. We use the names of these companies as targets for our analysis, and extract daily embeddings for them. Examples of selected companies include *Microsoft, Airbus, JPMorgan, Nintendo, Ford, Walmart...*

We use Yahoo Finance to extract the stock value time series for all 100 companies during the period under study. We keep only the closing price of each day, which is the value at the time the stock exchange closes every evening. As it is also closed on weekends and holidays, we have no value for these days. To align the time series of stock price with the daily embeddings series, we interpolate values for weekends and holidays, setting them to the average of the stock values of the preceding and following day.

Models. We apply the usual preprocessing steps such as removing stopwords and train a CBOW model on the whole corpus. Then, we use the incremental method described in Section 1.5 to fine-tune the model on each time slice and extract daily embeddings.

We also experiment with Temporal Referencing (see Section 1.4). We replace every entity e by its temporal counterpart e_t with the suffix t being the day when the entity appears. We then train a unique CBOW model on the whole corpus.

Finally, we use BERT with the standard extraction method, obtaining one embeddings for each occurrence of the name of a company. We apply two aggregation methods, averaging and clustering (see Section 2.2). For the averaging method, we compute the average embeddings of the entity at each time slice. For the clustering, we apply k-means on all embeddings of an entity, and deduce the distribution over the clusters at each time slice. We arbitrarily set the number of clusters to 7.

We can compare the embeddings at two different time slices with cosine distance (for CBOW embeddings and BERT with averaging) or Jensen-Shannon Divergence (for BERT with clustering). We compute time series of temporal drifts of embeddings using these distance measures. As in the previous chapters, we compute two types of drifts: the *incremental* drift, comparing each t with $t-1$, and the *reference* drift, comparing each t with a reference point *ref*.

[8] https://blog.aylien.com/free-coronavirus-news-dataset/

5.3.2 Correlations analysis

We compute the correlation between the time series of stock prices drifts, and two pieces of information extracted from the corpus: the time series of drifts of BERT and CBOW embeddings, and the time series of drift of the number of occurrences of the entities. All time series have around 200 points. *ref* is taken as the embedding trained on the full period (CBOW), the cluster distribution of the full period (BERT Clustering) or the average embedding on the whole period (BERT Averaging). Arguably, it would be more adequate to use an initial value at t_0 for comparison. However, this is a very domain-specific corpus; the data is too sparse at the beginning of the corpus to train robust embeddings on the first time slice, and embeddings pre-trained on another corpus would not be adequate. Thus, even though it leaks data from the future towards the past, we use the embeddings from the full corpus as reference. To match this choice of reference point for the embeddings, we use the average of the stock price time series and of the count time series over the full period as reference point.

Model	Comparison	t_i/ref	Mean correlation	t_{i+1}/t_i	Mean correlation
-	Stock / Count	0.23	0.489	0.36	0.34
CBOW	Stock	**0.57**	0.596	0.29	0.339
CBOW	Count	0.01	0.33	**0.98**	0.839
TR	Stock	0.24	0.434	0.18	0.334
TR	Count	0.30	0.406	0.37	0.372
BERT Avg	Stock	0.21	0.411	**0.64**	0.506
BERT Clust	Stock	0.22	0.439	0.21	0.439

Table 5.8 – Part of entities with significant Pearson correlation between embedding drift and Stock or Count drift, and average of their correlation.

Table 5.8 shows, for each pair of drift time series, the part of companies with significant Pearson correlation (p-value < 0.05) between the two series. We also provide the mean correlation. This is computed for the two types of drift: *incremental* and *reference*.

The movements of CBOW embeddings are highly correlated with the movement of the number of news from one day to the next one (*incremental*). This is coherent since, if the company is not mentioned in the news on a given day, its embeddings cannot drift. However, few entities have a significant correlation between their stock price movement and their count movement.

For CBOW, more than half entities have a significant correlation between their embeddings drift series and stock drift series (57%) from the reference point. The correlation is much lower using the incremental drift, as the day-to-day comparison is much noisier than the comparison with a reference point.

BERT embeddings suffer less from this noise, as the model does dot use the data of the period for fine-tuning like CBOW and TR. Thus, for the aver-

aging method, the incremental drifts of stock prices and embeddings are quite correlated (64 %). The clustering does not capture the same information, and fails to recognise a correlation with stock value drift. This might be due to the low context variability of entities, making the clustering struggle to find meaningful clusters of usage.

The Temporal Referencing method does not lead to correlated embeddings drift series; the amount of data at each time slice might be too low to capture high-quality representations for the entities.

5.3.3 Breakpoint detection

We observe a correlation between the reference drift of CBOW embeddings and stock prices, and between the incremental drift of BERT averaged embeddings and stock prices. But the correlation provides only limited information on the similarity of trend between the time series. When comparing time series, a common feature to observe are breakpoints. Breakpoint detection is a particularly important task in the financial domain, for application such as anomaly detection for fraud or market risk management. For every company, we compare the breakpoints of the time series of embeddings drifts with the breakpoints of the time series of stock prices drifts.

Given a signal, a breakpoint can be defined as a point where the signal changes his behaviour brutally. An overview of the literature on breakpoints detection is proposed by Truong, Oudre, and Vayatis (2020). To identify the K breakpoints in a signal y, we look for the optimal segmentation $\mathcal{S}$ of the signal into $K+1$ sub-signals by minimizing a criterion function $V(\mathcal{S}, y)$. This function is the sum of costs of all the sub-signals that make up the segmentation of the signal:

$$V(\mathcal{S}, y) = \sum_{k=0}^{K} c(y_{t_k:t_{k+1}}) \tag{5.1}$$

Where $y_{t_k:t_{k+1}}$ is the sub-signal between breakpoints k and $k+1$ and $c(\cdot)$ is the cost function. It captures the intra-sub-signal variability, by computing the distance between each point of the sub-signal and its empirical mean, for example using the L_2 norm. To successively detect the breakpoints, we use binary segmentation (Fryzlewicz, 2014). We first select the optimal breakpoint in the full signal and split the time series around this point. We repeat this method for each sub-signal until we reached the desired number of breakpoints.

We compare the detected breakpoints in the two series, by computing the proportion of "matching" breakpoints. We consider that breakpoints from two series are matching if they occur at the same time plus or minus a margin M. The justification behind this margin lies in the goal of the breakpoint detection. We assume that when an event happens in real life, it has an impact on the news and on the stock prices, but this might not be immediate. That is why

we allow this margin during the detection process, choosing $M = 2$ days.

The precision is the percentage of common breakpoints between the drifts series of stocks and embeddings. A precision of 1 means a perfect match between the two sets of breakpoints and a precision of 0 means no common breakpoints. The number of breakpoints has to be defined in advance. Too many breakpoints would artificially increase the precision; we choose $K = 7$ breakpoints (8 sub-signals), which gives the best trade-off between not being too sensitive to the stochasticity of the process and capturing a meaningful metric.

$$\begin{aligned}
&\text{Precision}: \mathbb{R}^K \times \mathbb{R}^K \to [0,1] \\
&\qquad (\text{Bkps}_{\text{emb}}, \text{Bkps}_{\text{Stock}}) \mapsto \frac{|C|}{K} \\
&\text{with C} = \{b_i \in \text{Bkps}_{\text{Emb}} \text{ s.t. } \exists b_j \in \text{Bkps}_{\text{Stock}} \text{ s.t. } |b_i - b_j| < M\}
\end{aligned}$$

Table 5.9 summarizes the results for two models (BERT with averaging and CBOW with incremental fine-tuning) and the two drift measures, *incremental* and *reference*. For each combination, we compute the average precision for all entities, and the proportion of entities with a precision strictly higher than $2/7$, corresponding to at least 3 matching breakpoints. To provide a point of comparison, we also compute the accuracy of a random baseline where breakpoints are uniformly sampled over the time steps.

The non-contextual embeddings CBOW detect slightly more stock price breakpoints, for a higher number of entities, compared to averaged BERT embeddings. Surprisingly, the breakpoint detection algorithm does not seem to suffer too much from the noise generated when computing the incremental drift: nearly half of the companies have at least 3 common breakpoints between their embedding drift time series and stock price drift time series.

Figures 5.7 and 5.8 show two examples of breakpoint detection with very high number of common breakpoints between the CBOW embeddings drift (bottom part of the figures) and stock price drift (top parts). The time series were computed using the *incremental* drift for two companies, HSBC and Blackrock. HSBC has a high significant Pearson correlation between the embeddings and stock drifts (0.5), while the correlation for Blackrock is not significant (0.07 with p-value > 0.05).

5.4 Conclusion

In this chapter, through two analyses, we showed how semantic variation can be linked with real-word events. First, interpreting word usage variation across time and between business sectors in companies annual reports allowed

Model	Measure	Average precision	Entities with precision $> 2/7$
CBOW	Reference	0.263	0.47
CBOW	Incremental	0.277	0.49
BERT	Reference	0.251	0.40
BERT	Incremental	0.276	0.32
random	-	0.155	0.08

Table 5.9 – Precision when detecting breakpoints for BERT and CBOW.

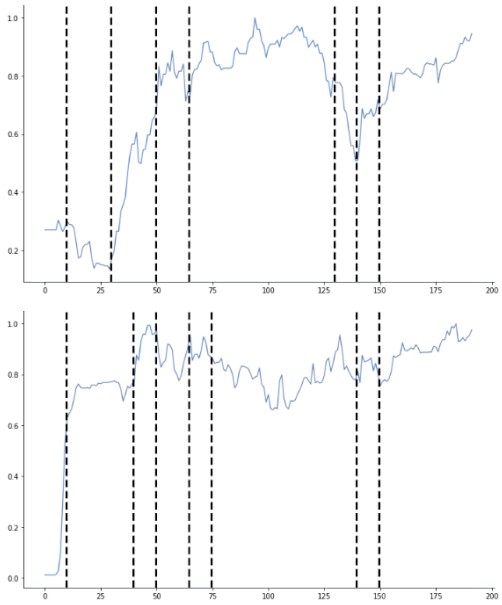


Figure 5.7 – Breakpoints for HSBC.

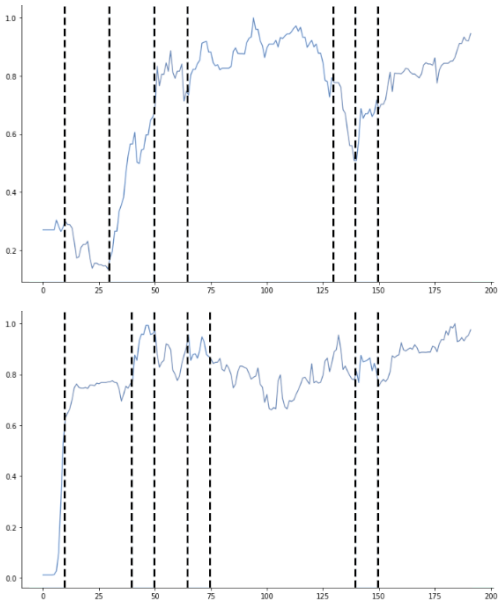


Figure 5.8 – Breakpoints for Blackrock.

us to understand the disparities in opinion and preoccupations between business sectors, and the evolution of these elements across time.

Second, the series of drifts of usage of company names in news articles could be linked with the evolution of the stock price of these companies. In particular, we could correlate these series for most companies. We showed that even for companies without correlation between embeddings and stock drifts, we could detect common breakpoints between these two series. However, additional experiments on time series prediction using LSTMs showed that using the time series of embeddings drifts as feature does not improve the accuracy of stock price prediction.

Conclusion

In this thesis, I explored methods to build time-varying word representations from a corpus in order to analyse their evolution. Many elements were written in collaboration with invaluable co-authors: Matej Martinc, from the Jozen Stefan Institute (Ljubjana, Slovenia), Elaine Zosa and Lidia Pivovarova, from the University of Helsinki (Finland), and Corentin Masson from LIMSI and University Paris-Saclay (France). Over the course of the PhD, I also collaborated with Aina Gari Soler, from LIMSI and University Paris-Saclay, and with Étienne Simon, from LIP6 and Sorbonne Université (France). I also spent a 5-months internship in the National Institute of Informatics in Tokyo (Japan) under the supervision of Pr. Asanobu Kitamoto. Finally, as part of my CIFRE contract with Société Générale, I contributed to the organisation of seminars and other NLP-related events in the company.

The main task I tackled is semantic change detection, defined as follows. A corpus of documents is first divided into time periods, leading to a set of pairs (text, time period). For a given target word, we extract a signal from all the contexts in which the word is used in each time period. This signal is a representation of the semantic information about the word at each period. Semantic change detection is done by characterizing the evolution of this representation across time. I focused on embedding-based methods, in particular contextualised embeddings extracted using pre-trained language models.

Summary of Contributions

There is a large body of existing work for semantic change detection. However, as in many domains, the models used in NLP have drastically changed in the last few years. Contextualized embeddings have become the standard for state of the art approaches. Starting my PhD in 2018, the literature associated with my research topic substantially evolved; this evolution can be seen through the experimental tools and contributions developed throughout this thesis.

Literature review on semantic change detection. The literature review proposed in Chapter 1 focuses on diachronic word representations using non-contextual embeddings. Most of the propositions can be classified in two broad types.

- Methods that consider time as a categorical variable: they learn word embeddings independently on each time slice and align them, or use methods such as *Temporal Referencing* or *Anchoring* to compare word representations from two separate time slices.

- Methods that consider time as an ordinal variable during training: "incremental" methods, relying on successive initialisation, and "joint" methods—also called "dynamic"—that use the information of all time slices simultaneously during training. We propose a set of experiments in Appendix C focusing on the ability of these methods to detect semantic change on scarce data.

Both types have their own limitations and drawbacks. For instance, alignment-based methods are source of noise and require a lot of data; incremental methods involve a lot of training updates for the embeddings; dynamic embeddings leak data from the future towards the past. More importantly, all these methods represent words as unique vectors in a time slice, without taking into account the possibility to have several distinct senses or usages. Some methods use word sense disambiguation or topic models to tackle word polysemy, but they can hardly be applied to a large vocabulary.

New perspectives can be found in the recent methods that learn contextualised word embeddings using pre-trained language models such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019). A short literature review on contextualised embeddings for diachrony is provided in the beginning of Chapter 2.

Finally, common corpora and evaluation methods used in the literature are listed in Chapter 3. Evaluation can be made in a qualitative fashion, by observing the evolution of some selected words with prior knowledge of their past meaning change. In a quantitative way, it is common to use indirect information, such as lexical databases and dictionaries, and downstream task such as word epoch disambiguation. It is also possible to use direct annotation of semantic change or semantic relatedness of words across time, but there are few of such datasets and they only cover small lists of words. Finally, to have a precise control of the drifts and the size of the evaluation corpus, many authors generate corpora with synthetic semantic drifts. These three strategies are used for evaluation in this thesis.

Extraction and aggregation methods using contextualised embeddings. In Chapter 2, we propose several methods to extract and aggre-

gate contextualised embeddings from pre-trained language models for semantic change detection.

First, we propose two methods to extract contextualised embeddings from BERT. The first one, the *standard* method, extracts and stores the embeddings of all occurrences of a word. We show the limitations of this method in terms of scalability, for example for situations where we want to identify semantic change in a large vocabulary. We propose a *scalable* method to extract only 200 embeddings per word per time slice, aggregating them as they are processed.

We also list four types of methods to quantify semantic change of a word across two periods, using *standard* or *scalable* extracted embeddings: Average Pairwise Distance, Averaging + Cosine Distance, Clustering + Jensen-Shannon Divergence, and Optimal Transport with Wasserstein or Sinkhorn divergence.

We apply them to several corpora in Chapter 3, comparing their behaviour and their ability to detect semantic change through two different evaluations: qualitative on a corpus of 20 time periods, and quantitative on annotated data from corpora with two time periods. We deduce the limitations of our different methods; among them, the one leading to the best accuracy on the annotated datasets relies on the Wasserstein distance to compare affinity propagation clusters of scalable BERT embeddings. Although it performs worse than a non-contextual embeddings method (Skip-Gram with Orthogonal Procrustes alignment), the clustering method offers a more fine-grained interpretation. It returns a distribution over a set of sentence clusters for each word in the corpus, that can be matched with word senses or usages. We apply this method to a corpus of COVID-related news articles, and link the distribution of word usages over time with real-word events. We show that clustering BERT embeddings and interpreting the clusters can enable analysts to highlight the impact of specific events and their echo across time.

Extension to several languages and dimensions. We extend the task of semantic change detection beyond the temporal dimension. First, in Chapter 4, we broaden the task of semantic change detection to a bilingual setting: we define and motivate the task of bilingual semantic change detection, as the study of the joint evolution of words in two corpora of different languages. We propose an experimental framework to characterise the relative drift of a word and its translation in another language, by defining several bilingual drift scenarios and generating a corpus of synthetic bilingual semantic change. We extend the diachronic methods using non-contextual and contextualised embeddings to build bilingual temporal word embeddings. The use of contextualised embeddings, coupled with a clustering step to aggregate the information from all the token embeddings of a word, leads to the best performance. The performance of the non-contextual embeddings, especially with independent training, is nevertheless competitive, which is coherent with the results of Chapter 3.

Then, Chapter 5 prolongs our semantic change detection methods to the

synchronic case. The detection of semantic variations can be performed across different sources or communities on top of time. We apply it to reports from companies in different business lines, highlighting how the usage of a word can vary across time and between various industries.

Applications. This thesis includes several examples of application of semantic change detection. In Chapter 3, we apply our scalable BERT embeddings extraction method to the Aylien corpus of COVID-related news. It allows us to rank all the words of a large vocabulary in terms of semantic change. Then, we apply clustering to the embeddings of the words to interpret the different senses and usages that they gained or lost over time. Thus, we demonstrate the large potential of contextual embeddings for the *interpretable* tracking of short-term changes in word usage, which has a practical application for crisis-related news monitoring.

Chapter 5 focuses on financial applications. Through the analysis of reports and statements from companies and regulatory authorities, we show that our semantic change detection and interpretation method can reveal the variations in preoccupations and opinions of different actors or sources. In particular, the ability to detect clear temporal tendencies in the distributions of word usage could allow a financial analyst to link these variations with real-world events, and have a deeper understanding of the phenomena behind them. Then, we compare the time series of usage variation of company names in news articles with the time series of their market value. We show that not only are they correlated, but we can also detect common breakpoints in these series, opening the way to joint analysis of financial indicators and semantic change information for many tasks in the financial domain.

Finally, in appendix D, we introduce the new task of detection of missing information in companies' financial reports. We propose to tackle it with an extension of our clustering and interpretation pipeline to this fraud detection task.

Future work

Limitations of the methods and evaluation. From our experiments in Chapter 3, we conclude that there is no such thing as a "best" semantic change detection approach. Different types of models are apt to tackle different aspects of the task; some models focus on the sequential aspects (non-contextual dynamic and incremental embedding models), others focus on context variation (contextualised embeddings with APD) or word sense disambiguation (clustering of contextualised embeddings). Moreover, the results show that contextualised embeddings are not necessarily the answer to everything, often being out-performed by non-contextual embeddings aligned across time periods. More importantly, when we rank the different semantic change detection

methods by accuracy, we observe that the best method differs depending on the evaluation corpus. This high variability can partly be explained through careful error analysis and by observing the models' behaviour; however, a much more thorough comparison is necessary to understand how these methods differ and which method is more fitting to what situation.

To this end, a precise control over evaluation is necessary. In particular, we would like to evaluate the different methods on various sub-tasks of semantic change. Which includes, among other tasks: detecting binary semantic change, measuring the speed of semantic change, detecting when a change happens, and identifying the type of semantic change (birth, death, broadening, narrowing...). Studying the impact of polysemy and frequency, and sensitivity to noise in general, is also crucial. All these elements can be controlled through the generation of synthetic semantic change. In line with Shoemark et al. (2019), a key future work is to build an extended evaluation framework for different aspects of semantic change detection. We would define and generate different types of scenarios to cover all the aforementioned situations and evaluate the different semantic change detection methods on the synthetic corpora.

Sequential aspect of semantic change. Due to the difficulty to annotate semantic change on corpora of more than two time slices, a large part of the related works evaluate their methods on corpora consisting of only two disjoint time periods: time is considered as a categorical nominal variable instead of ordinal. Few works treat time as ordinal, and even fewer consider it as a continuous variable (Rosenfeld & Erk, 2018).

Incremental and joint models, presented in Section 1.5, take the sequential information into account; independent models—including contextualised embeddings systems introduced in Chapter 2—deal with the different time periods of a corpus independently. In Section 3.2, we saw that these two categories of models do not always capture the same semantic shifts. Contextualised embeddings capture word polysemy and contextual variability in general; incremental or dynamic embeddings rely on word usage from the past to build embeddings for the present, highlighting smooth and directed drifts of word usages.

An important future work is to merge these two approaches, in order to include sequential information in contextualised embeddings. This can be done at different steps of the contextualised representation extraction. Here is a non-exhaustive list of possible strategies:

1. Instilling temporal information in the non-contextual word embeddings used by the pre-trained language model. Extracting contextualised word embeddings usually starts by associating non-contextual embeddings to the input tokens. This non-contextual representation can include external information. We could embed a temporal component, using any

non-contextualised diachronic embedding model from the literature. Using a dynamic model such as the one from Bamler and Mandt (2017) would provide some control over the drift of the word vectors.

2. Inserting this information inside the language model during training. For example, Peters, Neumann, Logan, et al. (2019) enhance BERT embeddings using information from a knowledge base.

3. Merging temporal information with the sequence embeddings at the output of the language model, as an additional layer. For example, Rosenfeld and Erk (2018) represent time through a time embedding and combine it with time-independent word embeddings to create diachronic embeddings.

4. Fine-tuning the language model incrementally in the successive time periods. This is also called "diachronic fine-tuning". Giulianelli, Fernandez, and Del Tredici (2019) and Kutuzov (2020) experiment with this strategy. Kutuzov (2020) incrementally fine-tunes ELMo models on 5 COHA decades, using the checkpoint of the model at the previous time slice to initialise the training on the next one. All decades were trimmed to the size of the smallest one to decrease the potential impact of frequency variation. He observes that this incremental fine-tuning leads to a spurious increase in lexical ambiguity at each time slice: as the model is further trained with new data, the diversity of contextualised embeddings increase. Moreover, they notice a higher correlation of word frequency with semantic change scores when using contextualised embeddings from incrementally fine-tuned language models. Finally, an evaluation on annotated data shows that the diachronically fine-tuned model under-performs the classical pre-trained model on semantic change detection. To sum up, this process has to be used with caution, as it is hard to differentiate real word usage variation from variation caused by the diachronic fine-tuning.

5. Learning temporal information through a downstream task, such as word epoch disambiguation, in addition to the language model objective. This can be done with a classification layer to predict the period of the sequence on top of a masked language model architecture.

Besides including temporal information in the contextualised embeddings, we wish to improve the way this information is modeled. A common method to model the time component in word embeddings is to use Gaussian processes (Bamler & Mandt, 2017; Rudolph & Blei, 2018). However, we could use different processes to model different possible behaviours of the word embeddings. In particular, in a short-term setting, we observe many breakpoints with sudden and sharp changes in word usage. For example, the word *diamond* that we study in Chapter 3 went through a sudden jump in its usage when the Diamond Princess cruise ship was quarantined in February 2020. To model this type of behaviour, we could use a jump process instead of a Gaussian

one. We could also use both types of process together, through the so-called jump-diffusion processes, as a prior on embeddings drifts. Common jump processes are Poisson processes. However, we would rather turn towards Hawkes processes, that are already extensively used in NLP for modelling information diffusion in social media (Mei & Eisner, 2017).

Applications. Automatic semantic shift detection has been used for text stream monitoring tasks, such as event detection (Kutuzov, Velldal, & Øvrelid, 2017) viewpoint analysis (Azarbonyad et al., 2017) or monitoring of rapid discourse changes during crisis events (Stewart et al., 2017). We believe that one of the key applications of short-term semantic change detection is news monitoring, and more specifically weak signal detection. A weak signal is a small piece of information that has ambiguous interpretation and implication, but may be of importance in the understanding of present events and in the prediction of future ones. Many anomaly detection tools are used on news and tweet streams to detect such information. However, few of them use word usage change as a weak signal.

Clustering of contextualised embeddings, as presented in the Chapter 3 use case, allows us to distinguish different types of word usage and track their distribution over time. To fully leverage the ability of this pipeline to detect and interpret word usage variations, our method can straightforwardly be extended in a streaming way. Any new document can be included in the analysis, be it a new central bank statement, company report, or in a classical streaming data situation such as daily financial news or tweets. Contextualised embeddings are extracted from the new document, and the clustering is updated using incremental clustering methods. For example, several incremental affinity propagation algorithms, designed for streaming data, are proposed in the literature (Ajithkumar & Wilson, 2017; Sun & Guo, 2014). Each new word embedding modifies the cluster distribution, either by being added to an existing cluster or by creating a new one.

List of publications

Martinc, M., Montariol, S., Zosa, E., & Pivovarova, L. (2020a). Capturing evolution in word usage: just add more clusters? In *Companion Proceedings of the Web Conference 2020* (pp. 343–349). WWW '20. Taipei, Taiwan: Association for Computing Machinery.

Martinc, M., Montariol, S., Zosa, E., & Pivovarova, L. (2020b). Discovery Team at SemEval-2020 Task 1: Context-sensitive embeddings not always better than static for semantic change detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation* (pp. 67–73). Barcelona.

Masson, C. & Montariol, S. (2020). Detecting omissions of risk factors in company annual reports. In *Proceedings of the Second Workshop on Financial Technology and Natural Language Processing* (pp. 15–21).

Montariol, S. & Allauzen, A. (2019a). Empirical study of diachronic word embeddings for scarce data. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)* (pp. 795–803). Varna, Bulgaria.

Montariol, S. & Allauzen, A. (2019b). Learning dynamic word embeddings with drift regularisation. *Conférence Nationale d'Intelligence Artificielle Année 2019*, 105.

Montariol, S. & Allauzen, A. (2020). Étude des variations sémantiques à travers plusieurs dimensions (Studying semantic variations through several dimensions). In *Actes de la 27e conférence Traitement Automatique des Langues Naturelles (TALN)* (pp. 314–322).

Montariol, S., Allauzen, A., & Kitamoto, A. (2020). Variations in word usage for the financial domain. In *Proceedings of the Second Workshop on Financial Technology and Natural Language Processing* (pp. 8–14).

Montariol, S., Martinc, M., & Pivovarova, L. (2021). Scalable and Interpretable Semantic Change Detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Montariol, S., Soler, A. G., & Allauzen, A. (2019). Exploring sentence informativeness. *Actes de la 26e conférence de Traitement Automatique des Langues Naturelles (TALN)*.

Appendices

Appendix A

Résumé en français

A.1 Introduction

Une langue est un système dynamique, en constante évolution dans le temps. Cette évolution est portée par la façon dont les gens utilisent les mots, et reflète l'évolution de divers aspects de la société tels que l'environnement technologique et culturel. Nous étudions les changements lexico-sémantiques: la variations temporelle dans l'usage et la signification des mots, également appelée *diachronie* en linguistique. Cette thèse explore et compare différentes méthodes permettant de construire des représentations de mots variant dans le temps à partir d'un corpus, pour analyser leur évolution.

Une motivation immédiate des méthodes de détection automatique du changement sémantique est de soutenir le travail des lexicographes, en analysant de grandes quantités de texte avec peu de moyens humains. Plus généralement, la détection des changements sémantiques peut être utilisée directement pour la recherche linguistique, sociologique et historique ; par l'interprétation des causes de ces changements, en les reliant à des événements réels, et en analysant l'évolution des tendances et des opinions au cours du temps (Gillani & Levy, 2019). Le changement sémantique peut également être utilisé comme outil pour de nombreuses tâches dans le traitement du langage naturel (TAL). De nos jours, un nombre croissant de données textuelles historiques sont numérisées et mises à la disposition du public. Elles peuvent être analysées en parallèle avec des documents contemporains, pour des tâches allant de la classification de textes à la recherche d'informations et la reconnaissance d'entités nommées (Jin et al., 2020).

Entre autres applications, cette thèse se concentre sur l'analyse de documents issus du domaine financier. En effet, il s'agit d'une thèse CIFRE, dans le cadre d'un partenariat entre le LISN-CNRS (ex-LIMSI), l'Université Paris-Saclay, et la Société Générale. Ce contexte ancre le thème de recherche dans des problématiques et enjeux concrets auxquels fait face l'entreprise. L'analyse de l'évolution sémantique dans le domaine financier s'inscrit en particulier dans

la gestion des risques. Qu'il s'agisse du risque de réputation, du risque politique ou du risque systémique, les variations de l'usage des mots peuvent fournir des informations précieuses aux analystes financiers.

Afin de détecter et d'analyser les changements lexico-sémantiques, la première étape consiste à extraire et à représenter la signification et l'usage d'un mot, à travers le contexte dans lequel il apparaît dans un corpus. Historiquement, les premiers travaux utilisent les fréquences relatives et les co-occurrences des mots (Sagi et al., 2009). Puis, les travaux de Bengio et al. (2003) et Mikolov, Sutskever, et al. (2013) proposent de représenter un mot pas un vecteur de réels : son plongement. Ce type de représentation prend en compte le contexte dans lequel le mot se trouve dans un corpus pour en déduire le vecteur qui lui est associé. Les différents usages du mot observés sur l'ensemble du corpus sont moyennés dans un vecteur unique. Pour étudier l'évolution de la signification des mots, nous avons besoin de plongements de mots variant dans le temps. Ils reposent sur l'extension suivante de l'hypothèse de distribution de Harris (1954) : un changement dans le contexte d'un mot reflète un changement dans sa signification ou son usage. Cette hypothèse conduit à l'émergence de plongements *diachroniques* de mots : les vecteurs de mots varient dans le temps, suivant l'apparition du mot dans des contextes différents.

Plus récemment, des modèles de langue pré-entraînés tels que BERT (Devlin et al., 2019) génèrent des plongements contextualisés : chaque occurrence d'un mot est représentée par un vecteur unique. Cela ouvre la voie à de nouvelles méthodes de détection de changement sémantique.

A.2 Contributions

Dans cette thèse, nous explorons, comparons et évaluons des méthodes de détection de changements sémantiques dans différents contextes.

La tâche principale abordée est la quantification du degré de changement sémantique, définie comme suit. Nous partons d'un corpus de documents, chacun associé à une date. Nous le divisons en plusieurs strates temporelles, selon la granularité choisie. Une strate peut être de n'importe quel ordre de grandeur, de quelques jours à plusieurs siècles, selon le corpus et le phénomène étudié. Nous obtenons ainsi un ensemble de paires (texte, période). Pour un mot cible donné, nous extrayons un signal de tous les contextes dans lesquels ce mot est utilisé dans chaque strate temporelle. Ce signal est une représentation de l'information sémantique du mot dans la strate. Nous étudions le changement sémantique comme l'évolution de cette représentation à travers le temps. Les méthodes que nous proposons visent à extraire ce signal.

La littérature sur la représentation des mots et le changement sémantique a évolué très rapidement pendant la préparation de cette thèse de doctorat. Le domaine du changement sémantique a connu une accélération soudaine depuis 2018, avec successivement l'émergence de modèles de plongements dynamiques;

la publication de plusieurs revues de littérature du domaine (Tahmasebi et al., 2018; Kutuzov et al., 2018); la tenue du premier workshop sur le changement sémantique (LChange19); et l'organisation d'une tâche d'évaluation (SemEval 2020 Task 1, Schlechtweg et al., 2020). En parallèle, tous les domaines du TAL ont été ébranlés par l'émergence des modèles de langue pré-entraînés tels que ELMo et BERT. Au cours de cette thèse, nous avons suivi cette évolution : elle se traduit par de grandes disparités entre nos premiers travaux et les plus récents. Dans l'ensemble, dans ce manuscrit nous nous concentrons sur les méthodes basées sur les plongements contextualisés extraits à l'aide de modèles de langue pré-entraînés.

A.2.1 Revue de littérature sur la détection des changements sémantiques

Avant l'émergence des plongements lexicaux, les premiers travaux de détection du changement sémantique se basaient sur le comptage des co-occurrences de mots. La revue de la littérature proposée dans le chapitre 1 se concentre sur les représentations diachroniques de mots utilisant des plongements non contextuels. La plupart des méthodes peuvent être classées dans deux catégories.

- Les méthodes qui considèrent le temps comme une variable catégorielle : elles entraînent les matrices de plongements de mots indépendamment sur chaque strate temporelle d'un corpus, puis les alignent — par exemple via une rotation dans l'espace vectoriel — ou utilisent des méthodes telles que *Temporal Referencing* ou *Anchoring* pour comparer les représentations des mots de deux périodes différentes (Kulkarni et al., 2015; Hamilton et al., 2016; Yin et al., 2018; Schlechtweg et al., 2019).

- Les méthodes qui considèrent le temps comme une variable ordinale pendant l'entraînement : cela inclut les méthodes dites "incrémentales", reposant sur une initialisation successive des modèles de plongements lexicaux avec les plongements de la période précédente; et les méthodes "jointes" — également appelées "dynamiques" — qui utilisent les informations de toutes les périodes simultanément pendant l'entraînement (Kim et al., 2014; Rudolph & Blei, 2018; Yao et al., 2018; Bamler & Mandt, 2017). Nous proposons un ensemble d'expériences dans l'annexe C qui se concentre sur la capacité de ces dernières méthodes à détecter les changements sémantiques sur des données en faible quantité.

Ces deux catégories de méthodes ont leurs propres limites et inconvénients. Par exemple, les méthodes basées sur l'alignement des espaces de représentation sont source de bruit et nécessitent beaucoup de données; les méthodes incrémentales impliquent de nombreuses mises à jour successives d'une matrice de plongements, induisant parfois des dérives fallacieuses dans les plongements;

enfin, les méthodes dynamiques utilisent les données de l'ensemble du corpus lors de l'apprentissage, prenant en compte des données des strates futures dans les plongements des strates passées. Plus important encore, toutes ces méthodes représentent chaque mot par un unique vecteur dans chaque strate temporelle, sans tenir compte de la possibilité d'avoir plusieurs sens ou usages distincts au sein d'une même période. Certaines méthodes utilisent la désambiguïsation du sens des mots ou des *topic models* pour aborder la polysémie des mots, mais elles peuvent difficilement être appliquées à un vocabulaire étendu (Mitra et al., 2015; Tahmasebi & Risse, 2017a; Frermann & Lapata, 2016).

De nouvelles perspectives peuvent être trouvées dans les méthodes récentes qui génèrent des plongements contextuels de mots en utilisant des modèles de langue pré-entraînés tels que ELMo (Peters et al., 2018) et BERT (Devlin et al., 2019). Des travaux récents, décrits au début du chapitre 2, montrent que les plongements contextuels peuvent être utilisés pour la détection des changements sémantiques en agrégeant les informations de l'ensemble des plongements d'un mot selon différentes méthodes (Martinc, Novak, & Pollak, 2020; Giulianelli et al., 2020; Kutuzov, 2020).

Enfin, les corpus communs et les méthodes d'évaluation utilisés dans la littérature sont énumérés dans le chapitre 3. L'évaluation peut être faite de manière qualitative, en observant l'évolution de certains mots pour lesquels on dispose de connaissances a priori sur leurs changements de sens au cours du temps. Pour l'évaluation d'un point de vue quantitatif, il est courant d'utiliser des informations indirectes, telles que des bases de données lexicales et des dictionnaires, et des tâches auxiliaires comme la classification de phrase dans différentes périodes. Il est également possible d'utiliser des corpus annotés en changement sémantique; mais de tels corpus sont peu nombreux et ne couvrent que des listes très réduites de mots (100 mots maximum). Pour finir, afin d'avoir un contrôle précis des dérives sémantiques et de la taille du corpus d'évaluation, de nombreux auteurs génèrent des corpus avec des dérives sémantiques synthétiques. Ces trois stratégies sont utilisées pour l'évaluation dans cette thèse.

A.2.2 Méthodes d'extraction et d'agrégation utilisant des plongements contextualisés

Les méthodes récentes d'apprentissage de plongements de mots contextualisés à l'aide de modèles de langue pré-entraînés comme BERT donnent de nouvelles perspectives au problème de la tâche de détection du changement sémantique. Cependant, elles impliquent une toute nouvelle façon d'entraîner, d'extraire et d'utiliser les plongements de mots.

Dans le chapitre 2, nous proposons plusieurs approches pour extraire et agréger les représentations contextualisées dans le temps, et quantifier le degré de changement sémantique des mots. Nous nous concentrons sur le passage à l'échelle de nos approches, en vue de les appliquer à de grands corpus ou à de

grandes listes de vocabulaire, et sur leur interprétabilité, en désambiguïsant les différents usages d'un mot au fil du temps. Nous évaluons l'efficacité de ces méthodes de manière qualitative et quantitative, en utilisant plusieurs corpus annotés. Nous les appliquons également de manière exploratoire à un corpus d'articles de journaux liés au COVID19 et proposons des interprétations des changements sémantiques détectés.

Tout d'abord, nous proposons deux méthodes pour extraire les plongements contextualisés de BERT. La première, la méthode *standard*, extrait et stocke les plongements de la totalité les occurrences d'un mot dans un corpus. Cette méthode est limitée en termes de passage à l'échelle, par exemple pour les situations où nous voulons quantifier le changement sémantique de tous les mots d'un large vocabulaire, ou pour quantifier le changement sémantique d'un mot très fréquent; en effet, le stockage et le calcul de la variation sémantique deviennent ardus pour de larges volumes de données. Nous proposons une méthode *scalable* pour extraire seulement 200 plongements par mot et par strate temporelle, en les agrégeant au fur et à mesure de leur extraction. Nous montrons qu'en plus de résoudre le problème de passage à l'échelle, cette méthode d'extraction permet de réduire le bruit dans le nuage de plongements, menant à une amélioration de la précision dans la tâche de détection de changement sémantique.

Nous listons ensuite quatre méthodes permettant de quantifier le changement sémantique d'un mot entre deux périodes, en utilisant les plongements extraits de façon *standard* ou *scalable* : distance moyenne par paire, distance entre les moyennes intra-période, clustering + divergence de Jensen-Shannon entre les distributions des clusters dans chaque période, et transport optimal avec divergence de Wasserstein et régularisation de Sinkhorn associés à un clustering.

Nous appliquons ces méthodes à plusieurs corpus dans le chapitre 3, en comparant leur comportement et leur capacité à détecter les changements sémantiques à travers deux évaluations différentes : qualitative sur un corpus de 20 strates temporelles, et quantitative sur des corpus annotés divisés en deux périodes. Nous en déduisons les limites de nos différentes méthodes. La méthode qui conduit à la meilleure précision sur les données annotées s'appuie sur la distance de Wasserstein pour comparer les clusters de plongements extraits de BERT avec la méthode *scalable*. Bien qu'elle soit moins performante qu'une méthode de plongements non contextuels (Skip-Gram avec alignement et distance cosinus), la méthode de clustering offre une interprétation plus fine. En effet, pour un mot donné, elle renvoie une distribution sur un ensemble de clusters de phrases. Ces clusters peuvent être mis en correspondance avec les différents sens et usages du mot; nous proposons une méthode d'interprétation basée sur l'extraction de mot-clés dans les clusters de phrases. Nous appliquons ce système à un corpus d'articles de presse liés au COVID19, et relions l'évolution de la distribution des usages des mots dans le temps avec des événements réels. Nous montrons que le partitionnement des plongements de BERT et l'interprétation des clusters peuvent permettre aux analystes de

mettre en évidence l'impact d'événements spécifiques sur l'usage des mots, et leur écho dans le temps.

A.2.3 Extension à plusieurs langues et dimensions

Nous étendons la tâche de détection des changements sémantiques au-delà de la dimension temporelle. Premièrement, dans le chapitre 4, nous élargissons la tâche de détection du changement sémantique à un cadre bilingue : nous définissons et motivons la tâche de détection du changement sémantique bilingue, comme l'étude de l'évolution conjointe d'un mot et de son équivalent dans deux corpus de langues différentes. Nous proposons un cadre expérimental pour caractériser la dérive relative d'un mot et de sa traduction dans une autre langue, en définissant plusieurs scénarios de dérive bilingue et en générant des corpus synthétiques où nous contrôlons le changement sémantique bilingue. Nous étendons les méthodes diachroniques utilisant des plongements non contextuels et contextualisés pour construire des plongements de mots à la fois bilingues et diachroniques. L'utilisation de plongements contextualisés, couplée à une étape de clustering pour agréger les informations provenant de tous les plongements d'un mot, permet d'obtenir les meilleures performances. La performance des plongements non contextuels, en particulier avec un entraînement indépendant sur les différentes strates temporelles, est néanmoins compétitive, ce qui est cohérent avec les résultats du chapitre 3.

Ensuite, le chapitre 5 prolonge nos méthodes de détection de changements sémantiques au cas synchronique. La détection des variations sémantiques peut être appliquée pour comparer des textes provenant de différentes sources ou communautés, en plus de leur répartition dans différentes périodes. Nous l'appliquons à des rapports financiers d'entreprises de différents secteurs d'activité, en soulignant comment l'usage d'un mot peut varier dans le temps et entre différents secteurs. Nous montrons ainsi que nos méthodes permettent de détecter et d'interpréter les variations dans l'utilisation d'un mot à travers plusieurs dimensions, et que ces variations peuvent être liées à des événements réels.

A.2.4 Applications

La détection automatique de variation sémantique a été utilisée pour des tâches de surveillance de flux de texte, telles que la détection d'événements (Kutuzov, Velldal, & Øvrelid, 2017) l'analyse de points de vue (Azarbonyad et al., 2017) ou la détection de changements rapides de discours lors de crises (Stewart et al., 2017). Nous pensons que l'une des principales applications de la détection des changements sémantiques sur de courtes périodes est la surveillance des actualités, et plus particulièrement la détection de signaux faibles dans les données journalistiques ou issues de réseaux sociaux. Un signal faible est une information dont l'interprétation et les implications sont am-

biguës, mais qui peut être importante pour la compréhension des événements présents et la prévision des événements futurs. De nombreux outils de détection d'anomalies sont utilisés sur les flux d'actualités et de tweets pour détecter ce type d'information. Cependant, peu d'entre eux utilisent le changement d'usage des mots comme potentiel signal faible.

Cette thèse comprend plusieurs exemples d'application de la détection de changement sémantique. Dans le chapitre 3, nous appliquons notre méthode d'extraction *scalable* des embeddings générés par BERT au corpus Aylien d'articles de journaux liés au COVID19. Cela nous permet de classer tous les mots d'un vaste vocabulaire en terme de changement sémantique. Ensuite, nous appliquons un clustering aux plongements des mots contextualisés, afin de distinguer différents types d'utilisation des mots, de suivre l'évolution de leur distribution dans le corpus, et de là, d'interpréter les différents sens et usages que ces mots ont acquis ou perdu au fil du temps. Ainsi, nous démontrons le grand potentiel des plongements contextuels pour le suivi *interprétable* des changements à court terme dans l'utilisation des mots, et son application pratique pour la surveillance des actualités liées aux crises.

Le chapitre 5 porte sur les applications financières. Par l'analyse de rapports d'entreprises et de déclarations d'autorités de régulation, nous montrons que notre méthode de détection et d'interprétation des changements sémantiques peut révéler les variations dans les préoccupations et les opinions des différents acteurs financiers. En particulier, la capacité à détecter des tendances temporelles claires dans les distributions des usages de mots pourrait permettre à un analyste financier de relier ces variations à des événements réels et d'avoir une compréhension plus approfondie des phénomènes sous-jacents. Ensuite, dans un corpus journalistique, nous comparons les séries temporelles des plongements diachroniques de noms d'entreprises avec les séries temporelles de leur valeur de marché. Nous montrons une corrélation entre ces séries, ainsi que la présence de points de rupture communs. Cela ouvre la voie à une analyse conjointe des indicateurs financiers et des informations issues du changements sémantiques pour de nombreuses tâches dans le domaine financier.

Enfin, dans l'annexe D, nous introduisons la nouvelle tâche de détection d'informations manquantes dans des rapports annuels financiers d'entreprises, dans le cadre de la détection de fraude. Nous proposons d'aborder cette tâche avec une extension de notre système de détection et d'interprétation des variations sémantiques.

A.3 Conclusion

D'après nos expériences au chapitre 3, nous concluons qu'il n'existe pas de "meilleure" approche de détection des changements sémantiques. Différents types de modèles sont aptes à aborder différents aspects de la tâche ; certains modèles se concentrent sur les aspects séquentiels (modèles de plongements

dynamiques et modèles incrémentaux non contextuels), d'autres se concentrent sur la variation du contexte (plongements contextuels avec distance par paire moyenne) ou la désambiguïsation du sens des mots (clustering de plongements contextuels). En outre, les résultats montrent que les plongements contextuels ne sont pas nécessairement la réponse à tout, étant souvent surpassés par les plongements non contextuels alignées entre deux périodes successives.

Plus important encore, lorsque nous classons les différentes méthodes de détection de changements sémantiques en fonction de leur précision, nous constatons que la meilleure méthode diffère selon le corpus d'évaluation. Cette grande variabilité peut en partie s'expliquer par une analyse minutieuse des erreurs et par l'observation du comportement des modèles ; toutefois, une comparaison beaucoup plus approfondie est nécessaire pour comprendre en quoi ces méthodes diffèrent et quelle méthode est la plus adaptée à chaque situation.

À cette fin, un contrôle précis de l'évaluation est nécessaire. En particulier, nous souhaitons évaluer les différentes méthodes sur diverses sous-tâches du changement sémantique. Cela comprend, entre autres, les tâches suivantes : détecter le changement sémantique binaire, mesurer la vitesse du changement sémantique, détecter quand un changement se produit et identifier le type de changement sémantique (apparition ou disparition d'un sens, élargissement, rétrécissement...). L'étude de l'impact de la polysémie et de la fréquence des mots, et de la sensibilité au bruit en général, est également cruciale. Tous ces éléments peuvent être contrôlés par la génération de changements sémantiques synthétiques dans un corpus. De façon similaire à Shoemark et al. (2019) et à nos expériences du chapitre 4, un travail futur essentiel consiste à construire un cadre d'évaluation étendu pour les différents aspects de la détection du changement sémantique. Nous définirions et générerions différents types de scénarios pour couvrir toutes les situations mentionnées ci-dessus et évaluerions les différentes méthodes de détection du changement sémantique sur les corpus synthétiques.

Appendix B

Additional experimental results

Tables B.2 and B.1 show the Spearman correlation between ground truth word rankings and rankings outputted by the various approaches relying on Optimal Transport. We use the different methods to compute the distance between embeddings at two time periods, for both standard and scalable embeddings. A selection of these results, analysed in details, can be found in Section 3.4.2.

Clustering	Measure	GEMS	SemEval				DURel	Avg
			English	German	Swedish	Latin		
Standard embeddings								
aff-prop	JSD	0.510	0.313	0.436	-0.026	0.467	0.542	0.374
aff-prop	WD	0.386	0.643	0.490	-0.100	0.246	0.456	0.354
aff-prop	Sinkhorn 10	0.178	0.485	0.270	-0.204	0.337	0.133	0.200
aff-prop	Sinkhorn 1	0.190	0.518	0.283	-0.179	0.362	0.084	0.210
aff-prop	Sinkhorn 0.1	-0.151	-0.450	-0.112	0.190	-0.487	-0.168	-0.196
aff-prop	Sinkhorn 0.01	-0.307	-0.513	-0.219	0.172	-0.438	0.114	-0.199
2× aff-prop	WD	0.392	0.419	0.377	0.233	0.327	0.379	0.355
2× aff-prop	Sinkhorn 10	0.126	0.308	0.315	0.187	0.230	-0.035	0.188
2× aff-prop	Sinkhorn 1	0.137	0.318	0.318	0.187	0.230	-0.023	0.195
2× aff-prop	Sinkhorn 0.1	0.255	0.362	0.328	0.207	0.241	-0.009	0.231
2× aff-prop	Sinkhorn 0.01	0.404	0.435	0.376	0.250	0.296	0.307	0.345
k-means 5	JSD	0.508	0.189	0.528	0.238	0.324	0.560	0.391
k-means 5	WD	0.334	0.347	0.408	0.248	0.470	0.618	0.404
k-means 5	Sinkhorn 10	0.089	0.559	0.350	0.280	0.426	0.286	0.332
k-means 5	Sinkhorn 1	0.098	0.544	0.350	0.272	0.432	0.286	0.330
k-means 5	Sinkhorn 0.1	0.186	0.365	0.356	0.126	0.432	0.109	0.262
k-means 5	Sinkhorn 0.01	0.243	0.259	0.404	0.094	0.476	0.440	0.319
2× k-means 5	WD	0.337	0.409	0.432	0.290	0.457	0.514	0.407
2× k-means 5	Sinkhorn 10	0.106	0.450	0.382	0.268	0.453	0.205	0.311
2× k-means 5	Sinkhorn 1	0.117	0.467	0.382	0.271	0.461	0.205	0.317
2× k-means 5	Sinkhorn 0.1	0.306	0.445	0.377	0.272	0.433	0.289	0.354
2× k-means 5	Sinkhorn 0.01	0.345	0.424	0.408	0.281	0.462	0.516	0.406
none	apd-cos	0.176	0.514	0.262	0.150	0.315	-0.035	0.230
none	apd-canb	0.314	0.486	0.397	0.163	0.290	-0.151	0.250
none	cos-avg	0.398	0.315	0.437	0.206	0.486	0.656	0.417

Table B.1 – Spearman Rank Correlation between system output rankings and ground truth rankings using standard embeddings.

Clustering	Measure	GEMS	SemEval				DURel	Avg
			English	German	Swedish	Latin		
Scalable embeddings								
aff-prop	JSD	0.394	0.371	0.498	0.012	0.346	0.512	0.355
aff-prop	WD	0.369	0.456	0.421	0.264	0.397	0.484	0.399
aff-prop	Sinkhorn 10	0.131	0.234	0.316	0.086	0.188	0.212	0.195
aff-prop	Sinkhorn 1	0.120	0.236	0.340	0.086	0.220	0.209	0.202
aff-prop	Sinkhorn 0.1	-0.136	-0.254	-0.240	-0.086	-0.126	-0.196	-0.173
aff-prop	Sinkhorn 0.01	-0.185	-0.420	-0.362	-0.058	-0.053	-0.089	-0.195
2× aff-prop	WD	0.380	0.412	0.457	0.190	0.426	0.530	0.399
2× aff-prop	Sinkhorn 10	0.202	0.497	0.411	0.179	0.453	0.391	0.356
2× aff-prop	Sinkhorn 1	0.214	0.497	0.408	0.173	0.458	0.391	0.357
2× aff-prop	Sinkhorn 0.1	0.335	0.451	0.418	0.174	0.443	0.474	0.383
2× aff-prop	Sinkhorn 0.01	0.385	0.401	0.459	0.199	0.414	0.514	0.395
2× aff-prop	WD -weighted	0.358	0.419	0.463	0.214	0.395	0.551	0.400
2× aff-prop	Sinkhorn 10-weighted	0.195	0.492	0.403	0.170	0.418	0.454	0.355
2× aff-prop	Sinkhorn 1-weighted	0.202	0.490	0.406	0.163	0.425	0.458	0.357
2× aff-prop	Sinkhorn 0.1-weighted	0.332	0.438	0.417	0.183	0.430	0.472	0.379
2× aff-prop	Sinkhorn 0.01-weighted	0.356	0.417	0.452	0.204	0.394	0.528	0.392
k-means 5	JSD	0.430	0.316	0.508	0.073	0.358	0.658	0.390
k-means 5	WD	0.372	0.360	0.514	0.316	0.360	0.607	0.430
k-means 5	Sinkhorn 10	0.218	0.532	0.432	0.265	0.465	0.468	0.397
k-means 5	Sinkhorn 1	0.228	0.521	0.434	0.265	0.459	0.468	0.396
k-means 5	Sinkhorn 0.1	0.303	0.340	0.455	0.083	0.473	0.547	0.367
k-means 5	Sinkhorn 0.01	0.321	0.245	0.485	0.112	0.455	0.539	0.360
2× k-means 5	WD	0.339	0.375	0.482	0.299	0.502	0.512	0.418
2× k-means 5	Sinkhorn 10	0.199	0.462	0.486	0.256	0.477	0.532	0.402
2× k-means 5	Sinkhorn 1	0.212	0.456	0.486	0.261	0.475	0.532	0.404
2× k-means 5	Sinkhorn 0.1	0.331	0.445	0.484	0.257	0.484	0.549	0.425
2× k-means 5	Sinkhorn 0.01	0.341	0.380	0.503	0.300	0.491	0.518	0.422
2× k-means 5	WD -weighted	0.323	0.368	0.465	0.302	0.490	0.518	0.411
2× k-means 5	Sinkhorn 10-weighted	0.172	0.462	0.466	0.248	0.454	0.516	0.386
2× k-means 5	Sinkhorn 1-weighted	0.180	0.461	0.466	0.248	0.454	0.516	0.387
2× k-means 5	Sinkhorn 0.1-weighted	0.309	0.447	0.477	0.257	0.452	0.516	0.410
2× k-means 5	Sinkhorn 0.01-weighted	0.324	0.366	0.495	0.283	0.497	0.500	0.411
none	apd-cos	0.268	0.504	0.287	0.219	0.324	0.249	0.308
none	apd-canb	0.328	0.483	0.479	0.226	0.304	0.288	0.351
none	cos-avg	0.403	0.319	0.554	0.193	0.481	0.598	0.425
none	WD	0.312	0.386	0.416	0.252	0.283	0.526	0.363
none	Sinkhorn 10	0.193	0.406	0.307	0.264	0.300	0.486	0.326
none	Sinkhorn 1	0.204	0.399	0.306	0.261	0.294	0.486	0.325
none	Sinkhorn 0.1	0.285	0.369	0.312	0.234	0.275	0.509	0.331
none	Sinkhorn 0.01	0.316	0.393	0.393	0.263	0.274	0.567	0.368

Table B.2 – Spearman Rank Correlation between system output rankings and ground truth rankings using scalable embeddings.

At the top of Table B.3 we overview previous work on the GEMS test set. To train the models, Gulordava and Baroni (2011) used GoogleBooks Ngrams, Frermann and Lapata (2016) used an extended COHA corpus, and Giulianelli et al. (2020) used a subcorpus of COHA, identical to the one used in our experiments. Table B.3 shows the Pearson and Spearman correlations between the models' outputs and the human-annotated drifts.

Method	**Pearson**	**Spearman**
Related work		
Gulordava and Baroni (2011)	0.386	-
Frermann and Lapata (2016)	-	0.377
Giulianelli, Del Tredici, and Fernández (2020)	-	0.276
Fine-tuned BERT, Standard embeddings		
Averaging	0.317	0.349
k-means, $k = 5$	0.539	0.508
Affinity propagation	**0.560**	**0.510**

Table B.3 – Spearman Rank Correlation between system output rankings and ground truth rankings on the GEMS dataset.

Appendix C

Dynamic embeddings and data scarcity

Contents

In this thesis, we saw how semantic change can be inferred from drifts of time-varying word embeddings. Detecting semantic change has numerous application, from supporting socio-linguistics analysis to improving accuracy of NLP tasks on temporal corpora. However, from time to time, temporal data may be too sparse to build robust word embeddings and to discriminate significant drifts from noise. In this work, we compare three models to learn diachronic word embeddings on scarce data: incremental updating of a Skip-Gram from Kim et al. (2014), dynamic filtering from Bamler and Mandt (2017), and dynamic Bernoulli embeddings from Rudolph and Blei (2018). All these methods were introduced in Chapter 1. In particular, we study the performance of different initialisation schemes and emphasise which characteristics of each model are more suited to data scarcity, relying on the distribution of detected drifts. Finally, we regularise the loss of these models to better adapt to scarce data.

This appendix is an adaptation of the work published under the title "Empirical Study of Diachronic Word Embeddings for Scarce Data" at RANLP 2019 (Montariol & Allauzen, 2019a).

C.1 Introduction

While many authors proposed diachronic embedding models these last years; word vectors varying through time, following the changes in the global context of the word. These methods usually need large amounts of data to ensure robustness. However, temporal datasets often face the problem of scarcity; beyond the usual scarcity problem of domain-specific corpora or low-resource languages, a temporal dataset can have too few data compared to the length of the period it covers. Moreover, the amount of digitised and publicly available historical texts is limited for many languages, particularly for oldest time periods.

This work addresses the following question: in case of scarce data, how to efficiently learn time-varying word embeddings? For this purpose, we compare three diachronic methods on corpora of different sizes. The first method is incremental updating (Section 1.5.1, Kim et al., 2014), where word vectors of one time step are initialised using the vectors of the previous time step. The second one is the dynamic filtering algorithm (Section 1.5.2, Bamler and Mandt, 2017) where the evolution of the embeddings from one time step to another is controlled using a Gaussian diffusion process. Finally, we experiment with dynamic Bernoulli embeddings (Section 1.5.3, Rudolph and Blei, 2018) where the vectors are jointly trained on all time slices.

The hyper-parameters of these models are specifically tuned towards efficiency on small datasets. Then, we explore the impact of different initialisation schemes and compare the behaviour of word drifts exhibited by the models. Finally, we experiment regularising the models in order to tackle the faults detected in the previous analysis. The experiment are made on the *New York Times Annotated Corpus* (NYT, Sandhaus, 2008).

C.2 Diachronic models

The problem of data scarcity for training word embeddings is not new. However, it has seldom been tackled from the point of view of diachrony before. We give a short overview of the literature of diachronic embeddings in the data scarcity setting before presenting the three models we chose for our analysis.

C.2.1 Diachronic word embeddings and data scarcity

In Chapter 1, we presented many methods to extract time-varying word representations from a corpus. Here, we focus on neural word embeddings methods such as Word2Vec. The two most common methods to train diachronic word embeddings are alignment of vector space between two time slices (Hamilton et al., 2016) and incremental fine-tuning of embeddings from one time slice to the next one (Kim et al., 2014).

The alignment method requires to train word embeddings from scratch at each period. It necessitates large amounts of data for each time step to prevent overfitting. Moreover, in the case of sparse data, these methods are sensitive to random noise, which is difficult to dissociate from legitimate semantic drifts. Indeed, Tahmasebi (2018) shows that low-frequency words have a much lower temporal stability than high-frequency ones.

In Tahmasebi et al. (2018), the authors explain that usual methods for diachronic embeddings training such as the two previously mentioned are ineffective for dealing with low-frequency words and hypothesise that a new set of methods, pooled under the name of *dynamic* models, may be more adapted. These methods use probabilistic models to learn time-varying word embeddings while controlling the drift of the word vectors using a Gaussian diffusion process. Bamler and Mandt (2017) uses Bayesian word embeddings, which makes the algorithm more robust when dealing with sparse data; while Rudolph and Blei (2018) relies on Bernoulli distributions to learn the dynamic embeddings jointly across all time slices, taking advantage of the full dataset.

Outside of the framework of diachrony, several attempts aim at improving or adapting word embeddings to low-volume corpora in the literature. It can involve morphological information (Luong, Socher, & Manning, 2013) derived from the character level (Santos & Zadrozny, 2014; Labeau, Löser, & Allauzen, 2015), and often make use of external resources: semantic lexicon (Faruqui et al., 2015), and pre-trained embeddings from larger corpora (Komiya & Shinnou, 2018). Without external resources, Herbelot and Baroni (2017) show that simple word embeddings such as Word2Vec can be used to learn representations even for very rare words, through the use of tailored hyperparameters. It notably involves a higher learning rate combined with a greedy training process. However, to our knowledge, no work has attempted to apply similar solutions to the problem of sparse data in temporal corpora, even thought this situation has been faced by many authors, often suffering from short time steps for social media data (Stewart et al., 2017; Bamler & Mandt, 2017; Kulkarni et al., 2015).

C.2.2 Models

This section briefly reminds the characteristics of the three models under study: the Skip-Gram incremental updating algorithm from Kim et al. (2014),

the dynamic filtering algorithm of Bamler and Mandt (2017), and the dynamic Bernoulli embeddings model from Rudolph and Blei (2018). They are presented with more details in Chapter 1. We consider a corpus divided into T time slices indiced by t. For each time step t, every word i is associated with two vectors $u_i^{(t)}$ (word vector) and $v_i^{(t)}$ (context vector).

Incremental Skip-Gram (ISG, Section 1.5.1). This algorithm relies on the skip-gram model estimated with negative sampling (SGNS, Mikolov, Sutskever, et al., 2013). For the first time slice, the matrices of word embeddings $U^(1)$ and $V^(1)$ are initialised using a Gaussian random noise $\mathcal{N}(0, 1)$ before being trained according to equation 1.2. Then, for each successive time slice, the embeddings are initialised with values of the previous time slice following the methodology of Kim et al. (2014). This way, the word vectors of each time step are all in the same vector space and directly comparable.

Dynamic Filtering of Skip-Gram (DSG, Section 1.5.2). This method relies on the Bayesian extension of the SGNS model described by Barkan (2017). Its goal is to control the drifts of the embeddings in order to enforce smoother trajectories. The main idea is to share information from one time step to another, allowing the embeddings to drift under the control of a diffusion process. A full description of this approach, denoted as the filtering model, can be found in Bamler and Mandt (2017).

Dynamic Bernoulli Embeddings (DBE, Section 1.5.3). The DBE models extends the *Exponential Family Embeddings* (EFE, Rudolph et al., 2016), a probabilistic generalisation of the *Continuous Bag-of-Words* (CBOW) model of Mikolov, Sutskever, et al. (2013). The main idea is that the model predicts the central word vector conditioned on its context vector following a Bernoulli distribution. The drift of the word vectors across time is controlled by a Gaussian process. A detailed description of the model can be found in Rudolph and Blei (2018).

C.3 Experimental results

The goal of this study is to compare the behaviour of these three algorithms on low-volume corpora. We evaluate their predictive power on different volumes of data to compare the impact of two initialisation methods, and analyse the behaviour of the drift of the embeddings.

C.3.1 Experimental setup

As stated before, in the case of temporal corpora, the volume of data per time step can be low for several reasons: the time steps are very short, or the corpus itself is small, for example with domain-specific corpora, low-resource languages, or old and low-resource periods.

We use the *New York Times Annotated Corpus* introduced in Section 3.1.1. It contains around 1 855 000 articles ranging from January 1st 1987 to June 19th 2007. We divide the corpus into $T = 20$ yearly time steps (the incomplete last year is not used in the analysis) and held out 10% of each time step for validation and testing. Then, we sample several subsets of the corpus: 50%, 10%, 5% and 1% of the training set. This way, we can compare the models on each subset to evaluate their ability to learn diachronic word embeddings with little data.

We remove stopwords and choose a vocabulary of the $V = 10$k most frequent words. Indeed, a small vocabulary is more adequate for sparse data in a temporal analysis in order to avoid having too many words missing from time steps. The total number of words in the corpus after preprocessing is around 38.5 million. It amounts to around 200k words per time step in the 10% subset of the corpus, thus only 20k in the 1% subset.

To tune the hyperparameters, we use the log-likelihood of positive examples $\mathcal{L}_{pos}$ measured on the validation set. We train each model for 100 epochs, with a learning rate of 0.1, using the Adam optimiser. For the DSG model, we use a diffusion constant $D = 1$ and a prior variance $D_0 = 0.1$ for both corpora. For the DBE model, we use $\lambda = 1$ and $\lambda_0 = 0.01$.

We choose an embedding dimension $d = 100$, as experiments show that a small embedding dimension, as Stewart et al. (2017) do, leads to smoother word drifts and makes the model less sensitive to noise when the data is scarce.

We use a context window of 4 words and a negative ratio of 1; we observed that having a higher number of negative samples artificially increased the held-out likelihood, but equalised the drifts of all the words in the corpus. Thus, in extreme scarcity situations, each negative sample has a high weight during training: the number of negative samples has to be very carefully selected depending on the amount of data.

C.3.2 Impact of initialisation on sparse data

The embedding vectors of the ISG and DBE models are initialised using a Gaussian white noise, while the means and variances of DSG are initialised with null vectors and identity matrices respectively. However, a good initialisation can greatly improve the quality of embeddings, particularly in the case of scarce data. We evaluate the impact of two types of initialisation on the log-likelihood of positive examples on the test set.

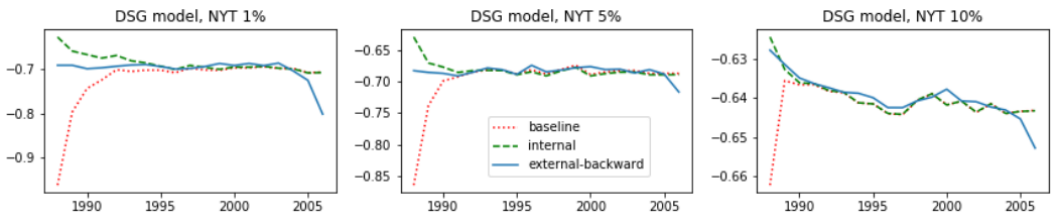


Figure C.1 – Log-likelihoods for the DSG model on three subsets of the corpus, comparing the baseline (random initialisation) with the two initialisation methods: *internal* is the initialisation from the full dataset while *external-backward* is the initialisation with the Wikipedia vectors, with training from most recent to oldest time step.

Internal initialisation: We train each model in a static way on the full dataset. Then, we use the resulting vectors as initialisation for the first time step of the diachronic models. This methods is especially suited to domain-specific corpora where no external comparable data is available.

Backward external initialisation: We use a set of embeddings pre-trained on a much larger corpus for initialisation: the Wikipedia corpus (dump of August 2013, Li et al., 2017) with vectors of size 100. These embeddings are representative of the use of words in 2013; and in general, large corpora exist almost exclusively for recent periods. Thus, we choose to use the pre-trained embeddings as initialisation for the *last* time step (the most recent). Then, we update the embeddings incrementally from new to old (*reverse incremental updating*). This method would be particularly suitable for corpora with low volume in older time slices, as it is the case for most historical datasets. For the DSG model, the pre-trained vectors are used as the mean parameter for each word. The variance parameter is fixed at 0.1. Experiments with a prior variance of 0.01 and 1 had a lowest log-likelihood on the validation set.

The log-likelihood curves in figure C.1 show that the internal initialisation has a better impact on the likelihood at the beginning of the period, as it is closer to the data than the external initialisation. The positive impact of the backward external initialisation increases with the volume of data.

Overall, the mean log-likelihoods across all time steps (Table C.1) are higher using the internal initialisation. We conjecture that internal initialisation is more profitable to the model when the period is short (here, two decades) with low variance. The backward external initialisation has very close scores to the internal one, and is more suitable for higher volume datasets on a longer period, as it improves the likelihood especially on bigger subsets.

Initialisation / Model	Random	Internal	Backward external
ISG	-3.17	-2.589	-2.686
DSG	-0.749	-0.686	-0.695
DBE	-2.935	-2.236	-2.459

Table C.1 – Log-likelihood on the 5% subset of the NYT corpus for each model, with the three initialisation schemes.

C.3.3 Visualising word drifts

A high log-likelihood performance does not necessarily imply that the drifts detected by the models are meaningful. In this section, we examine the distribution of word drifts outputted by each model with the internal initialisation. The computed drift is the L2-norm of the difference between the embeddings at t_0 and the embeddings at each t:

$$\text{drift}(U_i, t) = \left[\sum_{j=1}^{d} (u_{ij}^{(t)} - u_{ij}^{(t_0)})^2\right]^{1/2}. \tag{C.1}$$

This is similar to the inceptive drift introduced in Section 1.6. In the case of the DSG model were the words are represented as Gaussian distributions, we compute the difference of the mean vectors.

We repeat the experiment of Section 3.2.2.1 and plot the superimposed histograms of *successive* drifts (Figure C.2) from $t_0 = 1987$ to each successive time step, for all studied models. For example, on the histograms, the lightest colour curve represents the drift between $t_0 = 1987$ and $t = 2006$ and the darkest one is the drift between $t_0 = 1987$ and $t = 1988$.

A first crucial property is the *directed* aspect of the drifts: when the words progressively drift away from their initial representation in a directed fashion. On 10% of the dataset, the DBE model shows well this behaviour, with a very clear colour gradient. It is also the case for the other models on this subset. With 1% of the dataset on the contrary, the ISG model is unable to display a directed behaviour (no colour gradient), while the two other models do. This is justified by the use of the diffusion process to link the time steps in equations 1.6 and 1.10: it allows the DSG and DBE models to emphasise the directed fashion of drifts even when trained on scarce data.

The second property to highlight is the capacity of the models to discriminate words that drift from words that stay stable. From the human point of view, a majority of words has a stable meaning (Gulordava & Baroni, 2011); especially on a dataset covering only two decades like the NYT. The DBE model has a regularisation term (equation 1.11) to enforce this property, and a majority of words have a very low drift on the histogram. However, on 1%

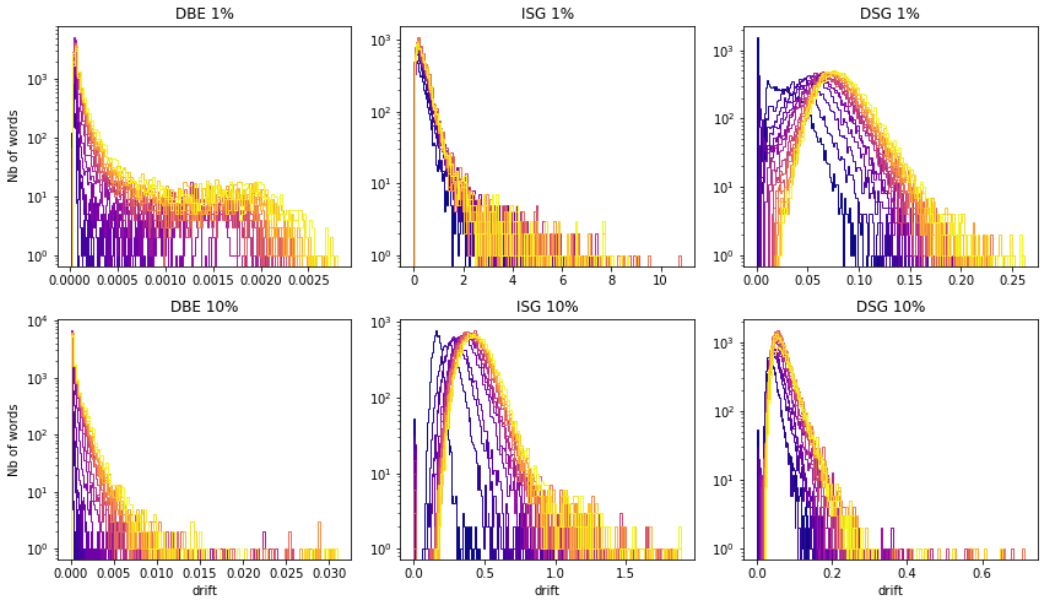


Figure C.2 – Histogram of word drift for each model on two subsets of the NYT corpus. The drifts are computed from $t_0 = 1987$ to each successive time step, and superposed on the histogram. The lightest colours indicate drifts calculated with the most recent time steps. The numbers of words are on a logarithmic scale.

of the dataset, this model cannot discriminate very high drifts from the rest. The ISG and DSG models have a different distribution shape, with the peak having a drift superior to zero.

To conclude, both the DBE and DSG models are able to detect directed drifts even in the 1% subset of the NYT corpus, while the ISG cannot. However, the drift distributions of the DBE and DSG models have a much shorter tail on the 1% subset than on the 10% subset: they are not able to discriminate very high drifts from the rest of the words in extreme scarcity situation.

C.3.4 Regularisation attempt

To tackle the weakness of the DBE and DSG models on the smallest subset, we attempt to regularise their loss in order to control the weights of the highest and lowest drifts. Our goal is to allow the model to:

- better discriminate very high drifts;
- be less sensitive to noise, giving lower weight to very low embedding drifts.

We test several possible regularisation terms to be added to the loss. The best result is obtained with the Hardshrink activation function, which is defined

this way:

$$\text{Hardshrink}(x) = \begin{cases} x & \text{when } x > \beta \\ -x & \text{when } x < -\beta \\ 0 & \text{otherwise} \end{cases} \tag{C.2}$$

For the DSG and DBE models, we add to the loss the following regularisation term, amounting to a thresholding function applied to the drift:

$$\text{reg}_\beta = \alpha \times \text{Hardshrink}\left(\sum_{i=1}^{V}\sum_{t=t_0}^{T} \text{drift}(U_i, t), \beta\right) \tag{C.3}$$

Where α is the regularisation constant to be tuned, β is the threshold of the hardshrink function, and the drift is computed according to equation C.1. This regularisation term is to be minimised. The activation function acts as a threshold to limit the amount of words having an large drift. We choose β as the mean drift for both models.

The drift histogram of the regularised embeddings can be found in Figure C.3. For both DSG and DBE, the right tail of the distribution of the drifts with regularisation is much longer than in the original model (Figure C.2). Moreover, in the case of the DSG model, more words have a drift very close to zero.

To conclude, the regularised DSG model considers more words as temporally stable. Furthermore, regularising the loss of the dynamic models allows them to better discriminate extreme word embedding drifts for very small corpora.

C.4 Summary & discussion

To summarise, we reviewed three algorithms for time-varying word embeddings: the incremental updating of a skip-gram with negative sampling (SGNS) from Kim et al. (2014) (ISG), the dynamic filtering applied to a Bayesian SGNS from Bamler and Mandt (2017) (DSG), and the dynamic Bernoulli embeddings model from Rudolph and Blei (2018) (DSG), a probabilistic version of the CBOW.

We proposed two initialisation schemes: the internal initialisation, more suited for low volume of data, and the backward external initialisation, more suited for higher volumes and long periods of temporal study. Then, we compared the distributions of the drifts of the models. We conclude that even in extreme scarcity situations, the DBE and DSG models can highlight directed drifts while the ISG model is too sensitive to noise. Moreover, the DBE model is best at keeping a majority of the words stable. This property, as well as the ability to detect directed drift, are two important properties of a diachronic model. However, both have low ability to discriminate the highest drifts on

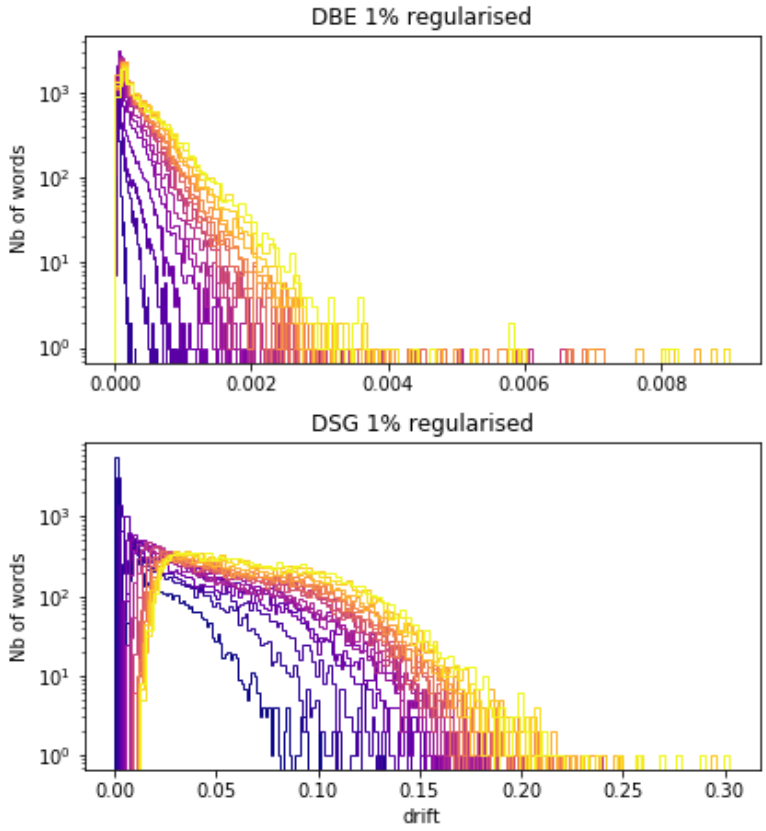


Figure C.3 – Histogram of word drift for the DBE and DSG regularised models on the 1 % subset.

a very small dataset. Thus, we added a regularisation term to their loss using the Hardshrink activation function, successfully getting longer distribution tails for the drifts.

To sum up, many tricks can be used to adapt non-contextual embeddings to scarce data situations; however, it might be out-performed by contextualised embeddings. Nowadays, pre-trained language models allow to extract contextualised word embeddings from a corpus whatever its size, as it does not require to be trained on it. Thus, it is well-suited for data scarcity situations. Even in the case of low-resource languages, multilingual language models achieve high-quality word representations for monolingual tasks (Conneau et al., 2020). Moreover, the aggregation of contextualised embeddings can be done at the level of the full corpus. For example, the clustering is done on the embeddings of all periods jointly, exploiting all available information; this is similar to the way dynamic embedding methods are trained on all time slices jointly.

Appendix D

Fraud detection: detecting omissions in financial reports

Contents

This work was done in collaboration with Corentin Masson from LIMSI and University Paris-Saclay, and published in Masson and Montariol (2020). We propose an extension of the sense clustering and interpretation method from Chapter 2 to a completely different setting. In the domain of fraud detection, we focus on detecting omissions of risk factors in companies annual reports. We extract sentences talking about risks and cluster their representations, and compute the distribution of these risk clusters for several dimensions (the year, the size of the company and the business line) to obtain a cartography of risks as a function of these dimensions.

D.1 Introduction

Risk analysis is a popular task in Business and Management research. While usually approached through expert knowledge and quantitative inputs (Kaplan & Garrick, 1981), it can benefit from the use of unstructured data

such as legal and regulatory documents. One of the associated tasks is the automatic extraction of risk sentences.

Theoretically, a risk can be defined as a hazard with a potential for damage to an entity. Its meaning differs from the notion of uncertainty; in the former, one is able to quantify precisely the probability of occurrence and its potential impacts (Altham, 1983). Therefore, a risk can be defined as a triplet composed of the potential event characterized as a risk, its quantitative counterparts such as the probability of occurrence, and its possible consequences (Kaplan & Garrick, 1981). Thus, risk evocations can be identified by a topic-oriented summarization system able to detect occurrences of these triplets from natural language written documents such as Annual Reports (ARs).

Listed companies are regulated by the Financial Market in which their value is most traded in, often inducing the obligation to regularly publish information documents. ARs are supposed to exhaustively describe a company's current well-being, perspectives and the risks it is facing. In France, nearly 190 ARs are released each year from CAC40, CAC60 and CAC90 indexes (the principal French stock indexes from Euronext.).

The literature on corporate annual reports analysis is plentiful in the financial research community. However, from the NLP perspective, research is more scarce and much more recent, while offering a wide range of applications from stock markets volatility prediction (Kogan, Levin, Routledge, Sagi, & Smith, 2009) to fraud detection. More details on ARs and literature on this topic can be found in Section 5.1.

To the best of our knowledge, few authors tackle risk sentences extraction from non-HTML indexed ARs (Liu, Liu, Wang, & Tsai, 2018); they often rely on XBRL -indexed 10-K filings to identify risk factors markers (Huang & Li, 2011). However, automatic analysis of such raw long documents can be beneficial for the Financial and Regulatory sectors. These documents represent the vast majority of ARs disclosed worldwide and are composed of an average of 3500 sentences with various sections and topics. As for now, little has been done on extracting specific sections from Annual Reports or indexing them. In this work, we focus on extracting and analysing the risk factors from these ARs.

In France, the financial market is regulated by the Financial Market Authority (AMF). In particular, disclosure of ARs depends on the "Code Monétaire et Financier" and on the "Doctrine."[1] Companies must release every year a report containing all the requested information. If an element that might be important for a potential investor is missing from an AR, the company runs the risk of being accused of voluntarily omitting information, which is a fraud under French law.

From the extracted risk sentences, it is therefore possible to identify the possible omission of a risk in an AR by comparing its risk distribution to

[1] AMF guidance for righteous behavior on the market.

other ARs from the same sector and year. Therefore, in this work, (1) we propose a new task for omitted risk factors detection from the DoRe Corpus (Masson & Paroubek, 2020), composed of European Companies ARs; and (2) we present a resolution method based on Neural Risk Sentences Extraction and Unsupervised Risk Factors Clustering. We hope to gather enough interest to further advance research on this domain.[2]

For investors and regulators, risk sections are important parts of ARs, as they contain information about the risks faced by the companies and how they handle it. Mandi, Chakrabarti, Patodia, Bhattacharya, and Mitra (2018) extract risk sentences from legal documents using Naive Bayes and Support Vector Machine on section embeddings. Dasgupta, Dey, Dey, and Saha (2016) explore project management reports from companies to extract and map risk sentences between causes and consequences, using hand-crafted features and multiple machine learning methods. Ekmekci, Hagerman, and Howald (2019) performed a multi-document extractive summarization on a news corpus for a risk mining task. As it has not yet been done, we experiment extractive summarization on risk extraction task in ARs.

Automatic text summarization is the task of producing a concise and fluent summary while preserving key information and overall meaning. In recent years, approaches to tackle this difficult and well-known NLP problem make use of increasingly complex algorithms ranging from dictionary-based approaches to Deep Learning techniques (Xiao & Carenini, 2019). The current research trend deviates from general summarization to topic-oriented summarization (Krishna & Srinivasan, 2018), targeting a specific subject in the document such as risks in ARs in our case.

Focusing on detecting risk factors in ARs, sentence clustering and topic modeling have been extensively used for this task in the literature (Zhu, Yang, & Moazeni, 2016; Chen, Rabbani, Gupta, & Zaki, 2017). The evaluation is mostly done using intrinsic measures and by looking at the clusters or topics manually. Only Huang and Li (2011) manually define 25 risk factor categories, relying on ARs from the Securities Exchange Commission.

D.2 Pipeline

We propose a pipeline for risk factors omission detection in ARs of CAC40 listed companies. Each PDF document is converted into textual format, divided into sentences and processed sequentially in the original document order to extract the ones talking about risk. Then, two approaches are implemented to identify classes of risks in risk-oriented summaries. Outputs are then used to compute divergence between ARs from the same sector or from the same year, and therefore identify the companies more likely to omit a risk.

[2]Please contact corentin.masson@limsi.fr for access to the corpus.

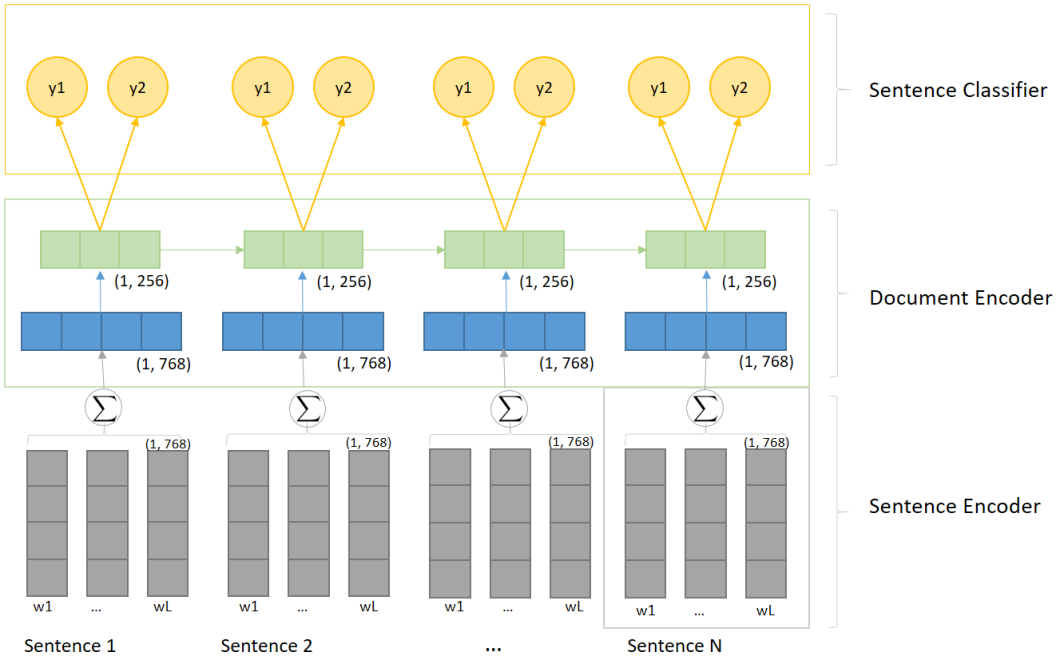


Figure D.1 – Risk Sentences Extraction architecture overview.

D.2.1 Risk sentences extraction

As presented in Figure D.1, each sentence in the document is processed sequentially using Flaubert (Le et al., 2020), a fine-tuned French version of BERT (Devlin et al., 2019). The goal is to compute the probability for each sentence to be a risk sentence using three modules: a Sentence Encoder, a Document Encoder and a Sentence Classifier.

D.2.1.1 Data description

ARs are often disclosed in PDF format, which requires a lot of pre-processing (a notable exception are 10-K filings, see Kogan et al., 2009). ARs are extremely long documents: they contain an average of 3500 sentences and 27 different sub-sections. Due to the large size of each document, completely labeling a set of reports would take a considerable amount of time. To handle this, we propose to split the document into a set of disjoint sub-documents and label by hand a randomly selected subset of these sub-documents.

D.2.1.2 Model architecture

The first module is a Sentence Encoder; its goal is to embed each sentence into a k-dimensional space without the information from the surrounding sentences. Due to the limited amount of labeled data, we use a FlauBERT pretrained Language Model and fine-tune it for the extraction task, allowing it

to get a good approximation of basic syntax and semantic features in higher layers (Jawahar et al., 2019).
With N_D being the number of sentences in a document $D = (S_1, S_2, \ldots, S_{N_D})$ and M_i being the length of the sentence $S_i = (w_1, w_2, \ldots, w_{M_i})$, SentEnc_i is the sum of the token embeddings computed by the fine-tuned FlauBert:

$$\text{SentEnc}_i = \sum_{j=1}^{M_i} \text{BERT}_{\text{TokenEmb}_j}(S_i)$$

We also experiment with a version where the sentence embeddings SentEnc_i are computed using the `[CLS]` token from the FlauBert model. In both cases, each sentence is mapped into a v dimensional vector.

Risk evocations are often split into multiple sentences. For example, in Figure D.2, the first sentence displays the risk factor while the second depicts the uncertainty with *"if"* and *"might"* along with the potential impact (*"affect its market share in a near future"*).

The sector is driven by innovation from newcomers. If the Group does not keep with the process, it might affect its market share in a near future.

Figure D.2 – Example of risk evocation.

We want our model to be able to extract all parts of the risk evocation. In order to extract sentence embedding taking into account the surrounding sentences (context sentences), we apply a forward LSTM layer at the document level, each sentence being considered as a token whose embedding comes from the Sentence Encoder. We take the hidden state of each sentence as the context sentence embedding.

$$\text{DocEnc}_i = \text{LSTM}(\text{SentEnc}_1, \text{SentEnc}_2, \ldots, \text{SentEnc}_{M_i})$$

As decoder, we add one linear layer with dropout for regularization. Its input comes directly from the contextualized sentence embeddings computed through the Document Encoder module, followed by a softmax layer to compute probabilities.

$$P(y_i = 1) = \text{Softmax}(\text{Linear}(\text{DocEnc}_1, \ldots, \text{DocEnc}_{N_D}))$$

For training, our loss function is a L2-penalized binary cross-entropy loss.

$$\mathcal{L} = -\sum_{d=1}^{N}\sum_{i=1}^{N_d}(y_i \log(p_i) + (1 - y_i)\log(1 - p_i)) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

D.2.1.3 Active learning

To our knowledge, there is no freely available dataset for risk sentences extraction in French nor in English, leaving us with a considerable labeling task. Randomly selecting sub-documents to label would be biased toward non-risk sentences and therefore would make the dataset asymmetric. Thus, we implement a Pool-Based Query-By-Committee (Settles, 2010) Active Learning approach using dropout masks for committee models generation and compute stochastic predictions for each sentence (Tsymbalov, Panov, & Shapeev, 2018). It allows to select the most informative sub-documents to label and increase the accuracy of the model for these sentences which are near the segmentation frontier.

With $L = \{D_1^L, D_2^L, \ldots, D_{N_L}^L\}$ the set of labeled sub-documents and $U = \{D_1^U, D_2^U, \ldots, D_{N_U}^U\}$ the set of unlabeled sub-documents, the framework – or Learner, as called in the Active Literature – looks for x^*, the most informative sentence with the selected query strategy. Our committee $H = \{h_1, h_2, \ldots, h_T\}$ is composed of T models. At each Active Learning iteration, a model is trained on the already labeled data. Then, T different dropout masks are applied on the classification layer of the Sentence Classifier module in order to generate T different model. They are used to compute stochastic predictions for each sentence in each sub-document.

Using the predictions for each sentence, we can compute the uncertainty score. As the Least Confidence, Sample Margin and Entropy measures are equivalent in the binary case, we compute the approximated Least Confidence measure using votes from the committee H for probability estimation p_i for each sentence. The uncertainty measure of a given sub-document is the average uncertainty score of all its sentences.

$$LS(D) = \frac{1}{N_D^U} \sum_{i=1}^{N_D^U} |p_i - 0.5|$$
$$\text{where } p_i = P(y_i = 1|X_i)$$

The learner ranks sub-documents by decreasing uncertainty measure and queries the M most informative sentences to the Oracle following : $x^* = \arg\max_{D^U} LS(D^U)$. The process is then iterated until a stop criterion is met, such as an insufficient increase of accuracy between two iterations.

D.2.2 Risk omission detection

We use the set of risk sentences extracted from the ARs to detect if a risk factor was omitted in a document.

D.2.2.1 Motivation & pipeline

All companies describe different types of risks in their ARs, often through a "risk factors" section. To detect if an AR is missing a risk factor that should have been reported, we would need to define a list of risks factors for all the companies. However, the regulators do not enforce any normalisation nor provide a list of risks to report. Thus, the number and the type or risks reported vary a lot in the different documents. Consequently, we have to use unsupervised methods to capture them.

From the sets of risk sentences, we create a mapping of the risks depending on the sector and the year of the ARs. The distribution of risks per year can also allow to identify emerging risks, while the distribution per sector allows to identify the risks that are specific to a sector. We can either work on the data at the sentence level using sentence clustering or at the document level by doing topic modeling. We present the two approaches in the following section.

D.2.2.2 Sentences clustering

This method is an extension of the one described in Section 2.2.4, but with clustering sentence representations instead of word representations. We cluster the risk sentences of all documents together to identify the types of risks across the full corpus. We use the sentence representations from the risk sentence extraction step using FlauBERT.

Moreover, we can assume that successive sentences, or sentences that are close in the document, have a high probability to deal with the same risk factor. Thus, the surrounding sentences as well as their distance to the target sentence can add valuable information to the clustering. We use the representation of the surrounding sentences as features for the clustering, by doing element-wise sum with the representation of the main sentence, weighted by a factor of their distance to the main sentence. The distance is computed according to the number of sentences: two successive sentences have a distance $d = 1$, etc. Then, the weight of each sentence is computed as the inverse of its distance to the main sentence augmented by one: $w = \frac{1}{d+1}$.

For the clustering, we use the k-means algorithm. The number of clusters k is chosen according to the literature on risk factors in ARs. To ease the interpretation of the different clusters of risk sentences, we use the method to detect keywords in the clusters described in Section 2.4.

D.2.2.3 Topic model on documents

We challenge the previous method using a popular topic modeling algorithm: the Latent Dirichlet Allocation (LDA, Blei et al., 2003). Each document is characterised by a probability distribution over a set of topics, while

each topic is characterised by a probability distribution over all the words of the vocabulary. Therefore, the top words per topic are used as a set of keywords to describe it. The number of topics is the same as the number of clusters for the sentence clustering with k-means.

D.2.2.4 Intrinsic evaluation measures

We compute several measures, all relying on a list of keywords characterising each topic or cluster.

First, the Normalized Point-wise Mutual Information (NPMI, Aletras and Stevenson, 2013) measures the topic coherence. It is introduced in Section 2.4. We also use external knowledge—pre-trained Word2Vec[3] embeddings (Mikolov, Sutskever, et al., 2013)—to evaluate topic coherence. Similarly to Ding, Nallapati, and Xiang (2018), we compute the pairwise cosine similarity between the vectors of the top k words characterizing each topic, and average it for all topics. We call this second topic coherence measure TC-W2V. For the two measures, we use a relatively low k ($k = 10$). A high NPMI or TC-W2V measure indicates an interpretable model.

These two measures are completed by a topic uniqueness (TU) measure (Nan et al., 2019) for the top k keywords, representing the diversity of the topics (see Section 2.4).

D.2.2.5 Risk omission detection task

The extrinsic evaluation is done using the detection of omissions as downstream task. We want to detect if a company omitted or under-reported a risk in one of its reports, by observing the risks reported in the document, and comparing it with the ones reported in other documents of the same year and the same sector.

First, we generate synthetic risk omissions in our corpus. We randomly sample a small set of ARs, manually select a section of each document describing one type of risk, and remove it. Our goal is double: to detect that a risk factor is missing in the altered document, and to identify the risk associated with the removed section.

To tackle this problem, we compute a measure relying on a binarized version of the topic distribution of a document. Indeed, both the topic model and the sentence clustering methods output a distribution of risks (respectively topics or cluster) for each document. We consider that a document includes a topic (or a cluster) if the proportion of the topic (or the number of sentences belonging to the cluster) is higher than a threshold ϵ. Below this threshold, we consider that the document does not report the risk characterised by that

[3]We use pre-trained French word embeddings on the Wikipedia Corpus: http://fauconnier.github.io

topic. Then, for each sector and for each year, we extract the set of "typical" topics: the ones that are present in most documents for that sector or year, and therefore are expected to appear in all documents of the same sector and year.

First, we count the number of documents mentioning each risk. Then, we binarize it: if the number of documents mentioning the risk is lower than half of the total number of documents in the sector/year, then the risk is considered as not important for the sector/year and we do not select it. We compare this list of "expected" topics with the list of topics reported in each document. It allows to identify the documents where a risk is absent but should have been reported, because it is a risk common to most documents for that sector or year.

For the second step, we check whether the missing topic detected by our method is the same as the one removed from the selected document. We use the fitted LDA and the fitted k-means algorithm to predict the topics (the clusters) which can be found in the set of sentences that were removed from the selected documents. If there is at least one topic in common between the set of "missing" topics in the document, and the set of topics predicted from the removed sections, we consider that the omission has been correctly detected.

In order to evaluate the ability of our methods to tackle the task, we define the accuracy measure as the proportion of correctly detected omissions among the 20 altered documents. This measure can be computed by using the documents of the same sector or of the same year as comparison; we name it *Binary-sector* and *Binary-year* accuracy. We also compute a joint measure, taking into account both the expected topics from the year and the ones from the sector: *Binary-all*.

D.3 Experiment

We use the pipeline described previously to extract the risk sentences with several iterations of labelling using active learning, and apply the aggregation methods to identify distinct risks factors. We use it to create a cartography of risks and deduce missing risk factors from artificially shortened documents.

D.3.1 Data preparation

D.3.1.1 Preparation for risk extraction

For labeling, we selected a random subset of 50 ARs from the whole DoRe Corpus containing French and Belgian companies with large, mid and small capitalization from various sectors. These documents are converted from PDF

to TXT format using MuPDF, [4] some were unusable and excluded after conversion, such as the 2018 AR from AIR LIQUIDE. We then extracted start and end offset of sentences from these documents using Stanza[5] from StanfordNLP team; we chose it for its accuracy and relative speed. All of these pre-processing steps induce errors; that is why we add some custom rules to filter out unusable sentences based on number of letters / sentence length ratios and counts of line-breaks in a sentence. To handle the cold start of our Active Learning approach, we label up to 1000 sentences in successive groups of 5 from the 4 first documents in the random sample. The labeling rule is to label a sentence as Risk sentence if it includes the notion of uncertainty, and if at least one other element from the Risk triplet is present. We take into account the surrounding sentences to check whether the missing element from the triplet is present in a sentence around the current one; if it is the case, we also label this second one as risk.

The initial set of 200 sub-documents is composed of groups of 5 successive sentences. We apply zero-padding to those with less than 5 sentences. We are unable to label a set of risk sentences representative of all potential risk topics from different sectors due to the size of the corpus; to evaluate the ability of the algorithm to detect risks even outside the sectors it has seen previously, we split the dataset into two parts and put sub-documents from two of the four first labeled ARs into the test set. This test set containing 70 sub-documents is used to follow the evolution of the performance metrics at each Active Learning iteration. It also allows the metrics during the Active Learning to be less sensitive to randomness of the split due to the low amount of data.

D.3.1.2 Active learning

From these selected data, we train the first model in our Active Learning pipeline. The parameters for our Query-By-Committee approach are the dropout probability of classification layers weights set to $p = 0.5$ and the number of models in the committee H set to $T = 15$ for computation feasibility.

We iterate 6 times and have 39% of risk sentences in the labeled sample. We can see in Table D.1 that the metrics globally increase during iterations while it is still subject to instability due to the lack of data. A solution to stabilize the results could be to add a cross-validation step, but it is computationally expensive.

D.3.1.3 Preprocessing for risk clustering

We focus on the CAC40 companies. We have 388 annual reports from 40 companies, spanning 12 sectors and 12 years (from 2008 to 2019). From the

[4] https://mupdf.com/
[5] https://stanfordnlp.github.io/stanza/

	Accuracy	F1	Recall
Iteration 1	0.8412	0.7373	0.7236
Iteration 2	0.8002	0.6403	0.6863
Iteration 3	0.8331	0.7483	0.6771
Iteration 4	0.8721	0.7767	0.8034
Iteration 5	0.8845	0.8158	0.7723
Iteration 6	**0.8969**	**0.8269**	**0.8216**

Table D.1 – Performance measures for each active learning iteration.

risk sentences extraction step, we have for each document, a set of risk-related sentences and their position in the document. On average, the extracted risk-related sentences correspond to 3.6% of the full document (minimum proportion = 1.3%, maximum = 14.1%). Each document is associated with a year and a company, which belongs to one of the 12 sectors. For both the topic modeling and the sentence clustering methods, the number of topics can be chosen by relying on the literature. Following Huang and Li (2011), we use $k = 25$ topics.

We apply a heavy processing step to all the risk sentences, in order to get a document as clean as possible to extract the most important keywords for each topic more efficiently. From the set of risk sentences, we first clean all errors resulting from the transition from pdf to text (divided words, merged characters...). Then, we exclude the sentences that have less than 60% of letters (too many symbols, spaces or digits in a sentence usually means that a portion of a data table was extracted). We delete numbers and symbols from the remaining sentences. We also remove French stopwords, words of less than 2 characters, words found in less than 15 documents and words found in more than 80% of the documents. Finally, we lemmatize all the words. [6]

D.3.2 Results

D.3.2.1 Risk sentence classification

We train two models for risk sentences classification, differing in the method to compute non-contextualized sentence embeddings. The first one (BERT Sum) is computed from the sum of the hidden-states of the last attention layer from the fine-tuned FlauBert model. The second model (BERT CLS) uses the CLS token, even though the extractive summarization literature tends to conclude that the second attempt is less accurate (Xiao & Carenini, 2019). Regarding the architecture, we set the Document Encoder LSTM hidden-states to 256, the Classifier Linear layer dropout probability to 0.5, the L2 penalization parameter of the loss function to 0.01 and the learning rate to 1.e÷5.

[6] For lemmatization, we use the `LefffLemmatizer()` from `Spacy`: https://pypi.org/project/spacy-lefff/

	Accuracy	F1	Precision	Recall
BERT CLS	0.8398	0.7679	**0.8968**	0.6715
BERT Sum	**0.8969**	**0.8269**	0.8323	**0.7723**

Table D.2 – Final results of both models after the final Active Learning iteration.

The model is optimized by Adam-Optimizer for 150 epochs with batch size of 16. We keep as best model the one having the best validation accuracy, and test it on the previously created test set (not used during Active Learning nor training).

Table D.2 presents the final results of both models after the last Active Learning iteration. Even if the (BERT CLS) Precision is better (0.8968), the increase in the recall (+0.1008) for (BERT Sum) makes it the best model for the task with the current amount of data. Table D.1 shows the results of the Active Learning step, increasing the F1 score by 0.0785 (10% increase in only 5 iterations). We believe that with a greater amount of data, the model can still increase its performance and gain a better capacity to identify unknown risk factors.

For each document, the risk sentences extracted by the model from each sub-document are concatenated to create the topic-oriented summary.

D.3.2.2 Risk clustering

In order to identify the different risk factors from the topic-oriented summary, we use the two unsupervised methods described before: sentence clustering and topic modeling.

On the one hand, we apply Online LDA (Hoffman, Bach, & Blei, 2010) to the set of risk sentences after preprocessing. On the other hand, we apply k-means to the set of sentence embeddings extracted from the Sentence Encoder. We experiment with k-means of sentences embeddings (KM), Augmented k-means using weighted embeddings of surrounding sentences with window = 2 (KM2), and Augmented k-means with window = 4 (KM4). As a preliminary measure of quality, we compute the silhouette score of the k-means clusterings. The score is the highest for the Augmented k-means with a window of 4 sentence (score = 0.178), slightly lower with a window of 2 sentences (score = 0.162), and even lower for the standard k-means (score = 0.147).

From the LDA, we have a set of keywords describing each topic. Some topic examples along with an interpretation of the associated risk factor are presented in Table D.5. To be able to compare it with the sentence clustering, we extract keywords from the sentence clusters from the k-means algorithm, using the tf-idf method (Section 2.4). Then, we compute the three intrinsic measures for both LDA and k-means to evaluate the quality of the topic model

	NPMI (k=10)	TC-W2V (k=10)	TU (k=25)
LDA	**-0.153**	0.175	**0.691**
KM	-0.240	**0.186**	0.652

Table D.3 – Intrinsic measures of topic modeling and sentence clustering quality.

and the clustering (Table D.3). The measures for the Augmented k-means are almost the same as for the standard k-means.

The measures show that the sentence clustering method leads to a higher extrinsic topic coherence (TC-W2V) than the topic model, but lower intrinsic topic coherence (NPMI). Moreover, the TU measure is lower for k-means, meaning that the clusters are less diversified.

D.3.2.3 Risk omission detection

We use the same models for the risk omission detection task. In order to generate synthetic omissions in ARs, we randomly sample and alter 20 ARs of the CAC40 companies, by manually removing a section describing one risk factor; and we add these altered documents to our corpus. We choose risk sections of different sizes, describing different types of risks; for example, we remove the *System security and cyber attack* section in the 2018 AR from ATOS, and the *Risk of delay and error in product deployment* section in the 2017 report from DASSAULT SYSTEMES.

After fitting the LDA and the k-means on the corpus, we obtain the distribution of risks in the altered documents and the average distribution of risks for each sector and year. According to the method described in Section D.2.2.5, we binarize these vector and compare them in order to identify the list of missing topics in the altered documents. Then, using the topic model and clustering fitted on the full corpus, we predict the distribution of risks in the sections that were removed from the selected documents. Finally, we can compute the accuracy measures described in Section D.2.2.5 using the LDA, the standard k-means and the Augmented k-means with windows of size 2 and 4 (Table D.4).

Augmenting the k-means algorithm by using the surrounding sentences, even though it improved the silhouette score, does not lead to a clear improvement for this task. However, the LDA leads to much lower accuracy compared to the k-means algorithm. It might be linked with the low extrinsic topic coherence of the LDA compared to k-means.

	LDA	KM	KM2	KM4
Binary - sector	0.2	0.7	**0.8**	**0.8**
Binary - year	0.2	**0.55**	0.4	0.4
Binary - all	0.4	0.75	**0.8**	**0.8**

Table D.4 – Accuracy measures for the risk omission detection task on the manually altered documents.

Risk factor	Example of keywords
reputation	agency, advertiser, publicity, affect, negatively
patent	property, intellectual, licence, brand, software
energy	oil, exploration, hydrocarbon, well, damage

Table D.5 – Translation of keywords examples using LDA with 25 topics, and manually associated risk factor.

D.4 Conclusion

In this work, we introduced the task of risk omission detection and proposed a pipeline to tackle it, relying on some methods introduced in Chapter 2. First, we extract risk sentences from company annual reports using an Encoder-Classifier architecture on top of contextualised embeddings from the BERT model. Then, we use unsupervised methods to extract the risk distribution of each annual report and interpret them.

We generate synthetic risk factor omissions in a sample of ARs in a straightforward way, propose a method to detect them, and a metric to evaluate the method. We conclude that a sentence-level analysis, by clustering sentence representation extracted with BERT, is more adapted than LDA to address the task. Augmenting the sentence clustering by using a weighted sum of the representations of the surroundings of a sentence can further increase its quality. The low performance of the LDA might be overcame using more advanced topic modelling methods (Nan et al., 2019), possibly relying on word embeddings.

However, the risk sentence extraction step could be improved with more Active Learning iterations, for the model to learn more about the notions of uncertainty and the impacts than about the risk factors that has already been observed during training. It could also be improved by increasing the number of sentences in each sub-document and transferring information between consecutive sub-documents in an AR.

References

Aitchison, J. (2001). Language change: progress or decay? In *Cambridge Approaches to Linguistics*. Cambridge University Press.

Ajithkumar, S. & Wilson, P. K. (2017). A survey paper on clustering data using incremental affinity propagation. In *IOSR Journal of Computer Engineering (IOSR-JCE)*.

Akbik, A., Blythe, D., & Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 1638–1649).

Alagić, D., Šnajder, J., & Padó, S. (2018). Leveraging lexical substitutes for unsupervised word sense induction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Aletras, N. & Stevenson, M. (2013). Evaluating topic coherence using distributional semantics. In *IWCS 2013* (pp. 13–22). Potsdam, Germany: ACL.

Altham, J. E. J. (1983). Ethics of Risk. *Proceedings of the Aristotelian Society*, *84*, 15–29.

Amrami, A. & Goldberg, Y. (2019). Towards better substitution-based word sense induction. *arXiv preprint arXiv:1905.12598*.

Azarbonyad, H., Dehghani, M., Beelen, K., Arkut, A., Marx, M., & Kamps, J. (2017). Words are malleable: computing semantic shifts in political and media discourse. (pp. 1509–1518).

Bamler, R. & Mandt, S. (2017). Dynamic Word Embeddings. In *Proceedings of the 34th International Conference on Machine Learning* (Vol. 70, pp. 380–389). Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR.

Barkan, O. (2017). Bayesian neural word embedding. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (pp. 3135–3143).

Basile, P., Caputo, A., Caselli, T., Cassotti, P., & Varvara, R. (2020). Overview of the EVALITA 2020 diachronic lexical semantics (DIACR-Ita) task. In *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*.

Beinborn, L. & Choenni, R. (2019). Semantic drift in multilingual representations. *Computational Linguistics*, 571–603.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. In *Journal of Machine Learning Research* (pp. 1137–1155).

Blank, A. & Koch, P. (1999). Why do new meanings occur? A cognitive typology of the motivations for lexical semantic change. In *Historical Semantics and Cognition* (pp. 61–89).

Blei, D., Kucukelbir, A., & McAuliffe, J. (2017). Variational inference: a review for statisticians. *CoRR*, *abs/1601.00670*.

Blei, D. & Lafferty, J. (2006). Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning* (pp. 113–120).

Blei, D., Ng, A., & Jordan, M. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Bloomfield, L. (1933). Language. In U. of Chicago Press. (Ed.).

Boberg, C. (2012). English as a minority language in Quebec. *World Englishes*, *31*.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Bréal, M. (1899). *Essai de sémantique (2nd ed.)* Hachette, Paris.

Buechel, S., Junker, S., Schlaak, T., Michelsen, C., & Hahn, U. (2019). A time series analysis of emotional loading in central bank statements. In *Proceedings of the Second Workshop on Economics and Natural Language Processing* (pp. 16–21). Hong Kong.

Chen, Y., Rabbani, M., Gupta, A., & Zaki, M. (2017). Comparative text analytics via topic modeling in banking. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1–8).

Church, K. W. & Hanks, P. (1989). Word association norms, mutual information, and lexicography. In *27th Annual Meeting of the Association for Computational Linguistics* (pp. 76–83). Vancouver, British Columbia, Canada.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 8440–8451).

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., & Jégou, H. (2018). Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Cook, P., Lau, J. H., McCarthy, D., & Baldwin, T. (2014). Novel word-sense identification. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (pp. 1624–1635). Dublin, Ireland.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems* (pp. 2292–2300).

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., & Salakhutdinov, R. (2019). Transformer-XL: Attentive Language Models beyond a Fixed-Length

Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 2978–2988). Florence, Italy.

Daille, B. (1994). Approche mixte pour l'extraction de terminologie : statistique lexicale et filtres linguistiques.

Dasgupta, T., Dey, L., Dey, P., & Saha, R. (2016). A framework for mining enterprise risk and risk factors from news documents. In *COLING 2016* (pp. 180–184). Osaka, Japan.

Daudert, T. & Ahmadi, S. (2019). CoFiF: A corpus of financial reports in French language. In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing* (pp. 21–26). Macao, China.

Davies, M. (2010). The Corpus of Contemporary American English as the first reliable monitor corpus of English. *Literary and linguistic computing*, *25*(4), 447–464.

Davies, M. (2012). Expanding horizons in historical linguistics with the 400-million word Corpus of Historical American English. *Corpora*, *7*(2), 121–157.

De Saussure, F. (1916). Cours de linguistique générale.

Del Tredici, M., Fernández, R., & Boleda, G. (2019). Short-term meaning shift: a distributional exploration. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 2069–2075). Minneapolis, Minnesota.

Delli Bovi, C., Camacho-Collados, J., Raganato, A., & Navigli, R. (2017). EuroSense: automatic harvesting of multilingual sense annotations from parallel text. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 594–600). Vancouver, Canada.

Desola, V., Hanna, K., & Nonis, P. (2019). FinBERT: pre-trained model on SEC filings for financial natural language tasks.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 4171–4186). Minneapolis, Minnesota.

Ding, R., Nallapati, R., & Xiang, B. (2018). Coherence-aware neural topic modeling. In *EMNLP 2018* (pp. 830–836). Brussels, Belgium: ACL.

Dorow, B. (2005). Using curvature and markov clustering in graphs for lexical acquisition and word sence discrimination. In *MEANING-2005, 2nd Workshop organized by the MEANING Project*.

Dubossarsky, H., Hengchen, S., Tahmasebi, N., & Schlechtweg, D. (2019). Time-Out: temporal referencing for robust modeling of lexical semantic change. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 457–470). Florence, Italy.

Dubossarsky, H., Weinshall, D., & Grossman, E. (2017). Outta control: laws of semantic change and inherent biases in word representation models.

In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 1136–1145). Copenhagen, Denmark.

Dyer, T., Lang, M., & Stice-Lawrence, L. (2017). The evolution of 10-K textual disclosure: Evidence from Latent Dirichlet Allocation. *Journal of Accounting and Economics*, *64*(2-3), 221–245.

Eger, S. & Mehler, A. (2016). On the linearity of semantic change: investigating meaning variation via dynamic graph models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 52–58). Berlin, Germany.

Ekmekci, B., Hagerman, E., & Howald, B. (2019). Specificity-based sentence ordering for multi-document extractive risk summarization.

Ethayarajh, K. (2019). How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *EMNLP/IJCNLP*.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., & Smith, N. A. (2015). Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1606–1615). Denver, Colorado.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Bradford Books.

Ferrari, A., Donati, B., & Gnesi, S. (2017). Detecting domain-specific ambiguities: an NLP approach based on Wikipedia crawling and word embeddings. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)* (pp. 393–399).

Frermann, L. & Lapata, M. (2016). A Bayesian model of diachronic meaning change. *Transactions of the Association for Computational Linguistics*, *4*, 31–45.

Frey, B. J. & Dueck, D. (2007). Clustering by passing messages between data points. *Science*, *315*(5814), 972–976.

Frossard, E., Coustaty, M., Doucet, A., Jatowt, A., & Hengchen, S. (2020). Dataset for temporal analysis of English-French cognates. In *Proceedings of The 12th Language Resources and Evaluation Conference* (pp. 855–859).

Fryzlewicz, P. (2014). Wild binary segmentation for multiple change-point detection. *Annals of Statistics*, *42*(6), 2243–2281.

Gillani, N. & Levy, R. (2019). Simple dynamic word embeddings for mapping perceptions in the public sphere. In *Proceedings of the Third Workshop on Natural Language Processing and Computational Social Science* (pp. 94–99). Minneapolis, Minnesota.

Giulianelli, M., Del Tredici, M., & Fernández, R. (2020). Analysing lexical semantic change with contextualised word representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 3960–3973).

Giulianelli, M., Fernandez, R., & Del Tredici, M. (2019). Contextualised word representations for lexical semantic change analysis. In *EurNLP*. London, UK.

Gonen, H., Jawahar, G., Seddah, D., & Goldberg, Y. (2020). Simple, interpretable and stable method for detecting words with usage change across corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 538–555).

Gulordava, K. & Baroni, M. (2011). A distributional similarity approach to the detection of semantic change in the Google Books Ngram corpus. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics* (pp. 67–71). Edinburgh, UK.

Hamilton, W. L., Leskovec, J., & Jurafsky, D. (2016). Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1489–1501). Berlin, Germany.

Han, R., Gill, M., Spirling, A., & Cho, K. (2018). Conditional word embedding and hypothesis testing via Bayes-by-Backprop. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 4890–4895). Brussels, Belgium.

Harris, Z. S. (1954). Distributional structure. *Word*, *10*(2-3), 146–162.

Herbelot, A. & Baroni, M. (2017). High-risk learning: acquiring new word vectors from tiny data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 304–309). Copenhagen, Denmark.

Hilpert, M. (2006). Distinctive collexeme analysis and diachrony. *Corpus Linguistics and Linguistic Theory*, *2*(2), 243–256.

Hoffman, M., Bach, F. R., & Blei, D. M. (2010). Online learning for latent dirichlet allocation. In *advances in neural information processing systems* (pp. 856–864).

Hovy, D. & Purschke, C. (2018). Capturing regional variation with distributed place representations and geographic retrofitting. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 4383–4394). Brussels, Belgium.

Howard, J. & Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 328–339). Melbourne, Australia.

Hu, R., Li, S., & Liang, S. (2019). Diachronic sense modeling with deep contextualized word embeddings: an ecological view. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 3899–3908). Florence, Italy.

Huang, K.-W. & Li, Z. (2011). A multilabel text classification algorithm for labeling risk factors in SEC form 10-K. *ACM Trans. Management Inf. Syst. 2*, 18.

Huang, X. & Paul, M. J. (2019). Neural temporality adaptation for document classification: diachronic word embeddings and domain adaptation models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 4113–4123). Florence, Italy.

Hussain, M. & Mahmud, I. (2019). pyMannKendall: a python package for non parametric Mann Kendall family of trend tests. *Journal of Open Source Software*, *4*(39), 1556.

Jatowt, A. & Duh, K. (2014). A framework for analyzing semantic change of words across time. *IEEE/ACM Joint Conference on Digital Libraries*, 229–238.

Jawahar, G., Sagot, B., & Seddah, D. (2019). What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 3651–3657). Florence, Italy.

Jawahar, G. & Seddah, D. (2019). Contextualized diachronic word representations. (pp. 35–47).

Jin, K., Wi, J., Kang, K., & Kim, Y. (2020). Korean historical documents analysis with improved dynamic word embedding. In *Applied Sciences 10(21)*.

Jordan, M. I., Ghahramani, Z., & et al. (1999). An introduction to variational methods for graphical models. In *Machine Learning* (pp. 183–233). MIT Press.

Kaji, N. & Kobayashi, H. (2017). Incremental skip-gram model with negative sampling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 363–371).

Kaplan, S. & Garrick, B. J. (1981). On the quantitative definition of risk. *Risk Analysis*, *1*(1), 11–27.

Kendall, M. (1975). Rank correlation measures.

Kilgarriff, A. (1997). I don't believe in word senses. *Computers and the Humanities*, *31*(2), 91–113.

Kim, Y., Chiu, Y.-I., Hanaki, K., Hegde, D., & Petrov, S. (2014). Temporal analysis of language through neural language models. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science* (pp. 61–65). Baltimore, MD, USA.

Kingma, D. P. & Welling, M. (2014). Auto-Encoding Variational Bayes. In *ICLR*.

Kogan, S., Levin, D., Routledge, B., Sagi, J., & Smith, N. (2009). Predicting risk from financial reports with regression. ACL.

Komiya, K. & Shinnou, H. (2018). Investigating effective parameters for fine-tuning of word embeddings using only a small corpus. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP* (pp. 60–67). Melbourne.

Krishna, K. & Srinivasan, B. V. (2018). Generating topic-oriented summaries using neural attention. In *the NAACL 2018* (pp. 1697–1705). New Orleans, Louisiana: ACL.

Kulkarni, V., Al-Rfou, R., Perozzi, B., & Skiena, S. (2015). Statistically significant detection of linguistic change. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 625–635). WWW '15. Florence, Italy: International World Wide Web Conferences Steering Committee.

Kutuzov, A. (2020). Distributional word embeddings in modeling diachronic semantic change. In *PhD Thesis*. University of Oslo.

Kutuzov, A. & Giulianelli, M. (2020). UiO-UvA at SemEval-2020 Task 1: Contextualised embeddings for lexical semantic change detection. *ArXiv*.

Kutuzov, A., Kuzmenko, E., & Pivovarova, L. (2017). Clustering of russian adjective-noun constructions using word embeddings. In *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing* (pp. 3–13).

Kutuzov, A., Øvrelid, L., Szymanski, T., & Velldal, E. (2018). Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 1384–1397). Santa Fe, New Mexico, USA.

Kutuzov, A., Velldal, E., & Øvrelid, L. (2017). Tracing armed conflicts with diachronic word embedding models. In *Proceedings of the Events and Stories in the News Workshop* (pp. 31–36). Vancouver, Canada.

Labeau, M., Löser, K., & Allauzen, A. (2015). Non-lexical neural architecture for fine-grained POS Tagging. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 232–237). Lisbon, Portugal.

Lance, G. N. & Williams, W. T. (1967). Mixed-data classificatory programs I.) Agglomerative Systems. *Australian Computer Journal*, 15–20.

Lau, J. H., Cook, P., McCarthy, D., Newman, D., & Baldwin, T. (2012). Word sense induction for novel sense detection. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 591–601). Avignon, France.

Le, H., Vial, L., Frej, J., Segonne, V., Coavoux, M., Lecouteux, B., Allauzen, A., Crabbé, B., Besacier, L., & Schwab, D. (2020). FlauBERT: unsupervised language model pre-training for French. In *Proceedings of The 12th Language Resources and Evaluation Conference* (pp. 2479–2490).

Levy, O. & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems* (pp. 2177–2185).

Lewis, C. & Young, S. (2019). Fad or future? Automated analysis of financial text and its implications for corporate reporting. *Accounting and Business Research*, *49*(5), 587–615.

Li, B., Liu, T., Zhao, Z., Tang, B., Drozd, A., Rogers, A., & Du, X. (2017). Investigating different syntactic context types and context representations for learning word embeddings. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 2411–2421).

Lin, J. (1991). Divergence Measures Based on the Shannon Entropy. *IEEE Trans. Inf. Theor. 37*(1), 145–151.

Lin, Z., Wan, X., & Guo, Z. (2019). Learning diachronic word embeddings with iterative stable information alignment. In *CCF International Conference on Natural Language Processing and Chinese Computing* (pp. 749–760). Springer.

Liu, Y.-W., Liu, L.-C., Wang, C.-J., & Tsai, M.-F. (2018). RiskFinder: a sentence-level risk detector for financial reports. In *NAACL 2018* (pp. 81–85). New Orleans, Louisiana.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE transactions on information theory*, *28*(2), 129–137.

Luong, T., Socher, R., & Manning, C. (2013). Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning* (pp. 104–113).

Mandi, J., Chakrabarti, D., Patodia, N., Bhattacharya, U., & Mitra, I. (2018). Use of artificial intelligence to analyse risk in legal documents for a better decision support. In *TENCON 2018*. Jeju, Korea (South).

Martinc, M., Novak, P. K., & Pollak, S. (2020). Leveraging contextual embeddings for detecting diachronic semantic shift. *LREC*.

Masson, C. & Paroubek, P. (2020). NLP analytics in finance with DoRe: a French 250M tokens corpus of corporate annual reports. In *LREC 2020* (pp. 2254–2260). Marseille, France.

Mei, H. & Eisner, J. M. (2017). The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems* (pp. 6754–6764).

Michel, J.-B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., Pickett, J. P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., et al. (2011). Quantitative analysis of culture using millions of digitized books. *science*, *331*(6014), 176–182.

Mihalcea, R. & Nastase, V. (2012). Word epoch disambiguation: finding how words change over time. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 259–263). Jeju Island, Korea.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 26* (pp. 3111–3119). Curran Associates, Inc.

Mitra, S., Mitra, R., Maity, S. K., Riedl, M., Biemann, C., Goyal, P., & Mukherjee, A. (2015). An automatic approach to identify word sense changes in text media across timescales. *Natural Language Engineering*, *21*, 773–798.

Mitra, S., Mitra, R., Riedl, M., Biemann, C., Mukherjee, A., & Goyal, P. (2014). That's sick dude!: Automatic identification of word sense change across different timescales. In *Proceedings of the 52nd Annual Meeting of*

the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1020–1029). Baltimore, Maryland.

Nan, F., Ding, R., Nallapati, R., & Xiang, B. (2019). Topic modeling with wasserstein autoencoders. In *ACL 2019* (pp. 6345–6381). Florence, Italy.

Navigli, R. & Ponzetto, S. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, *193*, 217–250.

Pasini, T. & Camacho-Collados, J. (2020). A short survey on sense-annotated corpora. In *Proceedings of The 12th Language Resources and Evaluation Conference* (pp. 5759–5765). Marseille, France.

Pechenick, E. A., Danforth, C. M., & Dodds, P. S. (2015). Characterizing the Google Books corpus: Strong limits to inferences of socio-cultural and linguistic evolution. *PloS one*, *10*(10).

Peng, H., Li, J., Song, Y., & Liu, Y. (2017). Incrementally learning the hierarchical softmax function for neural language models. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543).

Peters, M. E., Neumann, M., Logan, R., Schwartz, R., Joshi, V., Singh, S., & Smith, N. A. (2019). Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 43–54). Hong Kong, China.

Peters, M. E., Ruder, S., & Smith, N. A. (2019). To tune or not to tune? Adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)* (pp. 7–14). Florence, Italy.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of NAACL-HLT* (pp. 2227–2237).

Popescu, O. & Strapparava, C. (2015). Semeval 2015, task 7: Diachronic text evaluation. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* (pp. 870–878).

Purver, M., Valentincic, A., Pahor, M., & Pollak, S. (2018). Diachronic lexical changes in company reports: an initial investigation. In *Proceddings of the First Financial Narrative Processing Workshop (FNP 2018)*.

Ré, M. & Azad, R. (2014). Generalization of entropy based divergence measures for symbolic sequence analysis. *PloS one*, *9*.

Reif, E., Yuan, A., Wattenberg, M., Viegas, F. B., Coenen, A., Pearce, A., & Kim, B. (2019). Visualizing and measuring the geometry of BERT. In *Advances in Neural Information Processing Systems 32* (pp. 8594–8603).

Rodina, J., Bakshandaeva, D., Fomin, V., Kutuzov, A., Touileb, S., & Velldal, E. (2019). Measuring diachronic evolution of evaluative adjectives with word embeddings: the case for English, Norwegian, and Russian. In *Pro-*

ceedings of the 1st International Workshop on Computational Approaches to Historical Language Change (pp. 202–209). Florence, Italy.

Rodina, J., Trofimova, Y., Kutuzov, A., & Artemova, E. (2020). ELMo and BERT in semantic change detection for Russian.

Rong, X. (2014). Word2Vec parameter learning explained. *ArXiv*, *abs/1411.2738*.

Rosenfeld, A. & Erk, K. (2018). Deep neural models of semantic shift. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 474–484). New Orleans, Louisiana.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53–65.

Ruder, S., Vulić, I., & Søgaard, A. (2019). A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, *65*, 569–631.

Rudolph, M. & Blei, D. (2018). Dynamic embeddings for language evolution. In *Proceedings of the 2018 World Wide Web Conference* (pp. 1003–1011). WWW '18. Lyon, France.

Rudolph, M., Ruiz, F., Mandt, S., & Blei, D. (2016). Exponential family embeddings. In *Advances in Neural Information Processing Systems* (pp. 478–486).

Sagi, E., Kaufmann, S., & Clark, B. (2009). Semantic density analysis: comparing word meaning across time and phonetic space. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics* (pp. 104–111). Athens, Greece.

Sandhaus, E. (2008). The New York Times annotated corpus. In *Philadelphia : Linguistic Data Consortium*. Vol. 6, No. 12.

Santos, C. D. & Zadrozny, B. (2014). Learning character-level representations for part-of-speech tagging. In T. Jebara & E. P. Xing (Eds.), *Proceedings of the 31st International Conference on Machine Learning (ICML-14)* (pp. 1818–1826). JMLR Workshop and Conference Proceedings.

Schlechtweg, D., Hätty, A., Del Tredici, M., & Schulte im Walde, S. (2019). A wind of change: detecting and evaluating lexical semantic change across times and domains. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 732–746). Florence, Italy.

Schlechtweg, D., McGillivray, B., Hengchen, S., Dubossarsky, H., & Tahmasebi, N. (2020). SemEval-2020 Task 1: unsupervised lexical semantic change detection. *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Schlechtweg, D. & Schulte im Walde, S. (2020). Simulating lexical semantic change from sense-annotated data. *CoRR*, *abs/2001.03216*.

Schlechtweg, D., Schulte im Walde, S., & Eckmann, S. (2018). Diachronic usage relatedness (DURel): a framework for the annotation of lexical semantic change. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)* (pp. 169–174).

Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, *31*(1), 1–10.

Sen, P. K. (1968). Estimates of the regression coefficient based on Kendall's Tau. *Journal of the American Statistical Association*, *63*(324), 1379–1389.

Settles, B. (2010). *Active Learning Literature Survey*. University of Wisconsin, Madison.

Shoemark, P., Liza, F. F., Nguyen, D., Hale, S., & McGillivray, B. (2019). Room to glo: a systematic comparison of semantic change detection approaches with word embeddings. In *Proceedings of EMNLP-IJCNLP 2019* (pp. 66–76). Hong Kong, China.

Solomon, J. (2018). Optimal transport on discrete domains. *AMS Short Course on Discrete Differential Geometry*.

Stern, G. (1931). *Meaning and change of meaning; with special reference to the English language*. Wettergren & Kerbers.

Stewart, I., Arendt, D., Bell, E., & Volkova, S. (2017). Measuring, predicting and visualizing short-term change in word representation and usage in VKontakte social network. In *ICWSM*.

Sun, L. & Guo, C. (2014). Incremental affinity propagation clustering based on message passing. *Knowledge and Data Engineering, IEEE Transactions on*, *26*, 2731–2744.

Szymanski, T. (2017). Temporal word analogies: identifying lexical replacement with diachronic word embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (pp. 448–453). Vancouver, Canada.

Tahmasebi, N. (2018). A study on Word2Vec on a historical Swedish newspaper corpus. In *DHN*.

Tahmasebi, N., Borin, L., & Jatowt, A. (2018). Survey of computational approaches to diachronic conceptual change. *ArXiv*, *abs/1811.06278*.

Tahmasebi, N. & Risse, T. (2017a). Finding individual word sense changes and their delay in appearance. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017* (pp. 741–749).

Tahmasebi, N. & Risse, T. (2017b). Word sense change testset.

Tang, X. (2018). A state-of-the-art of semantic change computation. *Natural Language Engineering*, *24*(5), 649–676.

Tang, X., Qu, W., & Chen, X. (2016). Semantic change computation: A successive approach. *World Wide Web*, *19*(3), 375–415.

Taylor, W. A. (2000). Change-point analysis : a powerful new tool for detecting changes.

Tredici, M. D. & Fernández, R. (2017). Semantic variation in online communities of practice. In *IWCS 2017 - 12th International Conference on Computational Semantics - Long papers*.

Truong, C., Oudre, L., & Vayatis, N. (2020). Selective review of offline change point detection methods. *Signal Processing*, *167*, 107299.

Tsakalidis, A. & Liakata, M. (2020). Autoencoding word representations through time for semantic change detection. *ArXiv*, *abs/2004.13703*.

Tsakalidis, A., Bazzi, M., Cucuringu, M., Basile, P., & McGillivray, B. (2019). Mining the UK web archive for semantic change detection. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)* (pp. 1212–1221).

Tsymbalov, E., Panov, M., & Shapeev, A. (2018). Dropout-based active learning for regression. (pp. 247–258). Springer.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA*.

Vylomova, E., Murphy, S., & Haslam, N. (2019). Evaluation of semantic change of harm-related concepts in psychology. In *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change* (pp. 29–34). Florence, Italy.

Wiedemann, G., Remus, S., Chawla, A., & Biemann, C. (2019). Does BERT make any sense? Interpretable word sense disambiguation with contextualized embeddings. In *Proceedings of KONVENS 2019*. Erlangen, Germany.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xiao, W. & Carenini, G. (2019). Extractive summarization of long documents by combining global and local context. In *EMNLP-IJCNLP 2019* (pp. 3011–3021). Hong Kong, China: ACL.

Yao, Z., Sun, Y., Ding, W., Rao, N., & Xiong, H. (2018). Dynamic word embeddings for evolving semantic discovery. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (pp. 673–681). ACM.

Yin, Z., Sachidananda, V., & Prabhakar, B. (2018). The global anchor method for quantifying linguistic shifts and domain adaptation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (pp. 9434–9445). NIPS'18. Montréal, Canada.

Zhang, Y., Jatowt, A., Bhowmick, S., & Tanaka, K. (2015). Omnia mutantur, nihil interit: connecting past with present by finding corresponding terms across time. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 645–655). Beijing, China.

Zhu, X., Yang, S., & Moazeni, S. (2016). Firm risk identification through topic analysis of textual financial disclosures. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1–8).

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning Books and Movies: Towards Story-Like Visual

Explanations by Watching Movies and Reading Books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)* (pp. 19–27). ICCV '15. USA: IEEE Computer Society.

université PARIS-SACLAY | ÉCOLE DOCTORALE Sciences et technologies de l'information et de la communication (STIC)

Titre: Modèles diachroniques à base de plongements de mot pour l'analyse du changement sémantique

Mots clés: Plongements de mot, Diachronie, Changement sémantique, Modèles de langue, Multilingue, Domaine financier

Résumé:
Dans cette thèse, nous étudions les changements lexico-sémantiques : les variations temporelles dans l'usage et la signification des mots, également appelé *diachronie*. Ces changements reflètent l'évolution de divers aspects de la société tels que l'environnement technologique et culturel. Nous explorons et évaluons des méthodes de construction de plongements lexicaux variant dans le temps afin d'analyser l'évolution du langage. Nous utilisont notamment des plongements contextualisés à partir de modèles de langue pré-entraînés tels que BERT. Nous proposons plusieurs approches pour extraire et agréger les représentations contextualisées des mots dans le temps, et quantifier leur degré de changement sémantique. En particulier, nous abordons l'aspect pratique de ces systèmes: le passage à l'échelle de nos approches, en vue de les appliquer à de grands corpus ou de larges vocabulaire; leur interprétabilité, en désambiguïsant les différents usages d'un mot au cours du temps; et leur applicabilité à des problématiques concrètes, pour des documents liés au COVID19 et des corpus du domaine financier. Nous évaluons l'efficacité de ces méthodes de manière quantitative, en utilisant plusieurs corpus annotés, et de manière qualitative, en liant les variations détectées dans des corpus avec des événements de la vie réelle et des données numériques. Enfin, nous étendons la tâche de détection de changements sémantiques au-delà de la dimension temporelle. Nous l'adaptons à un cadre bilingue, pour étudier l'évolution conjointe d'un mot et sa traduction dans deux corpus de langues différentes; et à un cadre synchronique, pour détecter des variations sémantiques entre différentes sources ou communautés en plus de la variation temporelle.

Title: Models of diachronic semantic change using word embeddings

Keywords: Word embeddings, Diachrony, Semantic change, Language models, Multilingual, Financial domain

Abstract: In this thesis, we study lexical semantic change: temporal variations in the use and meaning of words, also called *diachrony*. These changes are carried by the way people use words, and mirror the evolution of various aspects of society such as its technological and cultural environment. We explore, compare and evaluate methods to build time-varying embeddings from a corpus in order to analyse language evolution. We focus on contextualised word embeddings using pre-trained language models such as BERT. We propose several approaches to extract and aggregate the contextualised representations of words over time, and quantify their level of semantic change. In particular, we address the practical aspect of these systems: the scalability of our approaches, with a view to applying them to large corpora or large vocabularies; their interpretability, by disambiguating the different uses of a word over time; and their applicability to concrete issues, for documents related to COVID19 and corpora of the financial domain. We evaluate the efficiency of these methods quantitatively using several annotated corpora, and qualitatively by linking the detected semantic variations with real-life events and numerical data. Finally, we extend the task of semantic change detection beyond the temporal dimension. We adapt it to a bilingual setting, to study the joint evolution of a word and its translation in two corpora of different languages; and to a synchronic frame, to detect semantic variations across different sources or communities on top of the temporal variation.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France