Détection d'évènements dans des flux de textes courts pour la prise de décision

Elliot Maître

SCALIAN

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *29/06/2022* **par :**

Elliot MAÎTRE

Event detection on streams of short texts for decision-making

---

JURY

Sylvie CALABRETTO	Professeure INSA LIRIS LYON	Rapporteur
Patrice BELLOT	Professeur Université Aix-Marseille	Rapporteur
Vincent CLAVEAU	Chercheur CNRS, Rennes	Examinateur
Cécile FAVRE	Maîtresse de conférences Université Lyon II	Examinatrice
Max CHEVALIER	Professeur, Université de Toulouse III	Directeur de thèse
Olivier TESTE	Professeur, Université de Toulouse II	Co-directeur de thèse
Bernard DOUSSET	Professeur, Université de Toulouse III	Co-directeur de thèse
Jean-Philippe GITTO	Docteur, Scalian	Encadrant Industriel

École doctorale et spécialité :
MITT : Domaine STIC : Intelligence artificielle
Unité de Recherche :
Institut de Recherche en Informatique de Toulouse (UMR 5505)
Directeur(s) de Thèse :
Max CHEVALIER, *Olivier TESTE* et *Bernard DOUSSET*
Rapporteurs :
Sylvie CALABRETTO et *Patrice BELLOT*

Remerciements

Je remercie dans un premier temps les deux structures qui m'ont permis de réaliser cette thèse. D'abord l'IRIT, pour le cadre académique fourni et Scalian pour le contexte industriel. Ce fut un réel plaisir d'évoluer à l'intersection de ces deux contextes.

Mes remerciements vont naturellement au jury qui a accepté d'évaluer mon travail. Plus spécifiquement, merci aux rapporteurs Patrice Bellot et Sylvie Calabretto pour leurs rapports positifs, détaillés et particulièrement constructifs à propos de mon travail, tout en pointant les limites de l'étude et surtout les axes d'amélioration. Merci Vincent Claveau et Cécile Favre d'avoir accepté d'être examinateur et examinatrice, et pour les discussions enrichissantes que nous avons pu avoir au cours de la soutenance et ensuite. Ensuite, merci à mes encadrants de Scalian, Zakaria et Jean-Philippe. Merci JP pour ta présence au quotidien, ton soutien et tes conseils, en tant que collègue, mais aussi en tant qu'ami. Merci à mes encadrants académiques, Max pour ta rigueur et ta volonté, qui m'ont poussé à être persévérant. Olivier, pour ton pragmatisme, ta bonne humeur, sympathie et ta patience, qui m'ont notamment aidé à corriger mes travers de langue. Enfin, merci Bernard pour ta bonne humeur et ta disponibilité.

Ces trois années de thèses se sont bien déroulées entre autres grâce à tous mes collègues, que ce soient les collègues de bureau à l'IRIT : Oihana, Clément, Inès, Nabil, Tianyi, Michele... Ou mes autres collègues de SIG : Daria, Nathalie, Omar, les Philippe... Merci tout le monde. Merci aussi aux collègues de chez Scalian, en particulier l'équipe lab : Daniel, Julien, Antoine, Clara, plus récemment Cyril, Manon, Aziz ainsi que les Parisiens et les Sophiapolitains. La team choco restera la meilleure aux CovidR ! Merci aussi aussi à la team enviro, notamment Xavier et Pepa, à Agathe, Claudina, et tous les autres collègues de Scalian côtoyés pendant ces trois ans. Un remerciement particulier pour Jean et Giovanna, qui m'ont beaucoup aidé sur la partie applicative de ce projet.

Dans un contexte un peu moins professionnel, les potes de la Martin (Léo, Antoine V, Antoine L, Yoann, Aurel, Quentin, Seb, ...) et de Télécom, qui ont suivi la thèse de (plus ou moins) loin pendant ces 3 ans. Toujours un plaisir de vous revoir quand c'est possible ! Je tenais à remercier les différentes personnes rencontrées à Toulouse. Anaïs, pour une rencontre des plus improbables, Margaux, pour toutes ces randos, discussions et plus généralement bons moments passés ensemble. Merci évidemment

à la team de l'A7 (François, P2, Sylv, Floky, Antho, Thomas, Paolo, Flo Vazzo...) et en particulier Robin, qui cumule plusieurs casquettes, pour leur accueil parmi eux. Je me suis tout de suite senti l'un des vôtres ! Et bien sûr merci aux copines respectives :) Merci aussi aux colocs du 3610, que ce soit les occupant.e.s du moment Martoche, Gégé, Paul, ou les ancien.nes Lucie, Jeanne, Simon, Tita. Ça a été un plaisir de vivre avec vous (et d'aller au POC) !

Viennent ensuite les Revermontois et assimilés, qui sont là depuis bien des années. Avec un système de détection d'événements comme celui-là, s'il se passe quelque chose de louche en Bresse, au moins on devrait être au courant rapidement. Merci Quentin, Thibal, Benj, Loïc, Corentin, Marlène pour toutes ces soirées et bons moments ensemble. Merci Manu, Perico, Ant, Art, Roxor pour les fous-rires, les chargeurs, les aventures ... Et vivement les prochaines.

Merci Léa pour ton accompagnement et ton amour ces derniers mois. Ton soutien a été vraiment précieux pour moi au quotidien, tu m'as permis de rester concentré sur mon objectif au cours de ces temps difficile en étant toujours présente pour moi. Merci aussi à ma famille, oncles et tantes, mais aussi cousins et cousines, qui visiblement ont dû m'inspirer par leur parcours ! Merci à Francis et Sophie pour leur bonne humeur, notre complicité, les rigolades et les grandes discussions pour les trop peu nombreuses fois où nous arrivons à nous voir ! Merci à mes parents pour tout le soutien, l'amour, la confiance qu'ils m'ont donnés au cours de mes études. Cette thèse est aussi votre réussite !

Enfin, je tenais à dédier cette thèse à mes grands-parents, qui se sont (un peu trop littéralement) tués à la tâche pour que leurs enfants et leurs petits-enfants aient la possibilité de pouvoir faire les études qu'ils n'ont pas eu la chance de faire. Merci pour cela, vous pouvez être fiers.

Abstract

The objective of this thesis is to conceive and build an event detection system on social networks to assist people in charge of decision making in industrial contexts. The event detection system must be able to detect both targeted, domain-specific events and general events. In particular, we are interested in the application of this system to supply chains and more specifically those related to raw materials. The challenges are to conceive such a detection system, but also to determine which events are potentially influencing the raw materials supply chains. This synthesis summarizes the different stages of research conducted to answer these problems.

First, we introduce the different modules of an event detection system. These systems are classically composed of a data filtering and cleaning step, ensuring the quality of the data processed by the system. Then, these data are embedded in such a way that they can be clustered by similarity. Once these data clusters are created, they are analyzed in order to know if the documents constituting them discuss an event or not. Finally the evolution of these events is tracked. In this thesis, we have proposed to study the problems specific to each of these modules.

We compared different text representation models, in the context of our event detection system (EDS). We also compared the performances of our event detection system to the First Story Detection (FSD) algorithm, an algorithm with the same objectives. We first concluded that our proposed system performs better than FSD, but also that recent neural network architectures perform better than TF-IDF in our context, contrary to what was shown in the context of FSD. We then proposed to combine different textual representations in order to jointly exploit their strengths.

Then, we have proposed different approaches for event detection and event tracking. In particular, we use Entropy and User Diversity to evaluate whether the clusters are related to events. We then track the evolution of event clusters over time by making comparisons between event clusters at different times, in order to create chains of event clusters. Lastly, we studied how to evaluate event detection systems in contexts where only few human-annotated data are available. We proposed a method to automatically evaluate event detection systems by exploiting partially annotated data.

In a final section, in order to specify the types of events to supervise, we conducted a historical study of events that have impacted the price of raw materials. In particular,

we focused on phosphate, a strategic raw material. We studied the different factors having an influence, proposed a reproducible method that can be applied to other raw materials or other fields. Finally, we drew up a list of elements to supervise to enable experts to anticipate price variations.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Decision-making is an increasingly difficult task in a world that becomes more and more complex. One must consider a large amount of information in order to make appropriate decisions. This massive information can take a different form depending on the source considered. Currently, content in the form of feeds is being democratized, whether for information (continuous news channels, newsfeeds)[1], for entertainment (streaming video games, chats)[2] or on social networks[3] where content is now offered in the form of an infinitely scrollable feed. These sources are designed to keep the listener's attention as long as possible so they contain a tremendous amount of data, in which one can find important information as well as mundane conversations, advertisement, or content designed for entertainment. While these feeds contain information about nearly everything that is happening around the world, extracting all this information is hardly doable by a human as it is time-consuming. Hence, one could be interested in an automated way to extract important information from these feeds, which is not a trivial task for a human nor for a machine.

1.1 Background and Objectives

1.1.1 Background

This thesis was conducted in collaboration with the R&D Laboratory of Scalian. The role of the R&D Laboratory is to conduct research and make innovative propositions about the expertise domains of the company to broaden its offer. Scalian is a consulting company working among others with buyers and supply chain managers in numerous domains such as aeronautics, transportation, energy and banking.

In order to manufacture and deliver its products to customers on time and with a satisfactory level of quality, a company must be able to manage its network of suppliers

[1] https://www.reuters.com/news/archive
[2] https://www.twitch.tv/
[3] https://twitter.com/

efficiently and proactively. Indeed, the capacity and capability of suppliers are critical elements that can heavily affect the company's production. This management is even more complicated when the supplier network is extensive and the types of flows are heterogeneous.

In the case of companies that manufacture complex systems (such as aerospace), the supplier network can incorporate thousands or even tens of thousands of actors on several levels. In this context, we talk about extended supply chains. In this case, specific methods and tools are required to manage the supply chain. If today there are ways to evaluate suppliers, select them, and manage them on precise perimeters, there is no efficient way to manage the end-to-end supply chain of a company.

Due to these difficulties, a vital component in corporate decision-making, whether global or local, is the information system, which enables the anticipation of risks, particularly those related to the supply chain. The information system contains the "internal" information needed to decide, act, learn, understand, forecast and control the functioning of the company and its various components. Other information, just as important to complete the vision of managers but less used, corresponds to data "external to the company". This data concerning the company's environment is under-exploited today.

Many of Scalian's industrial customers are in this extended supply chain configuration. Hence, Scalian is interested in developing a tool which allows the detection and anticipation of disruption of large supply chains. Internal data from clients is important but is not sufficient and must be completed by external data, particularly for suppliers that are far in the supply chain for which few information are accessible. Thus, exploiting external data such as feeds is a strategic task for the company in order to mitigate the risks associated with these disruptions. Specifically, we are interested in discovering events that can impact and disrupt supply chains, as early as possible, to take informed decisions and mitigate the impact of these events.

1.1.2 Objectives

Various data streams exist and they contain different types of data, such as text, images or videos. To extract important information from these streams, it is necessary to detect the moments of particular interest. According to (Champagne, 2000), "*societies undergo permanent structural transformations, which continuously produce facts bearing major consequences, immediate or future. The life of groups and institutions seems to be influenced by an uninterrupted succession of what we call "events". This intuition is based on the fact that some moments are indisputably stronger than others like rupture or reconciliation moments. These facts that are by nature non-ordinary draw the social attention of large fractions of the people and might last in the collective memory as major facts.*" Thus, events are the entities that convey the highest amount

of information, they are the cause of the disruptions that are happening. According to (Nora, 1972), "*Press, radio, images are not only means from which events are relatively independent; they are the very condition of their existence.*" The events depend on the media sharing it and their existence depend on that. As we are interested in detecting possible disruptions in supply chains, we want to detect these events, because they are by nature the most important things happening and hence are carrying the major part of the information. **In the rest of this thesis, we will tackle the problem of event detection on data streams.** Specifically, we chose to focus on textual data streams for several reasons:

- Potential sources are multiples, such as social networks and newsfeeds. As (Nora, 1972) stated, the diversity of the sources is a characteristic of an event.
- The production of textual data is one of the least time consuming, allowing a very good reactivity. Most of the real-time sources are in form of short texts, such as headlines or microblog.
- The amount of information carried by text data is high by nature.

In this study, we are interested in an automated way to detect events happening all around the world that might cause direct or future disruption of the supply chains. Different types of events can have such impact, from really general like the 2021 Suez canal obstruction or the Covid-19 pandemic breakout to new trading policies, new environmental rules, conflicts such as the war in Ukraine or specific like the fire of a warehouse of an important supplier or in an important commercial area, or technological breakthrough. We want to ease the process of decision-making by providing meaningful descriptions of the events. Figures 1.1 show a few examples of impacting events that were discussed online.

As we stated before, Scalian's customers are from various domains, thus talking about supply chain is a very broad term. Each type of product comes from a supply chain of varying size and composition. However, each product requires fundamental elements to make it, called raw materials. Scalian is particularly interested in conducting a study applicable to any type of domain and which could interest several of their clients. Thus, we propose an application of our work to the field of raw materials.

The scope of the study is the following : First, we want to build an event detection system to detect events in real-time using publicly available data stream, in an open domain manner, meaning that the event we are looking for are not specified. Indeed, we know that some events are causing disruptions, however we cannot explicitly specify which one to look for as most of them are not known beforehand. This first part is the core of the thesis and most of the contributions are related to the event detection system. Then, we want to establish which events can cause a disruption in the raw materials supply chain, which events cause the variation of the stock prices, and what

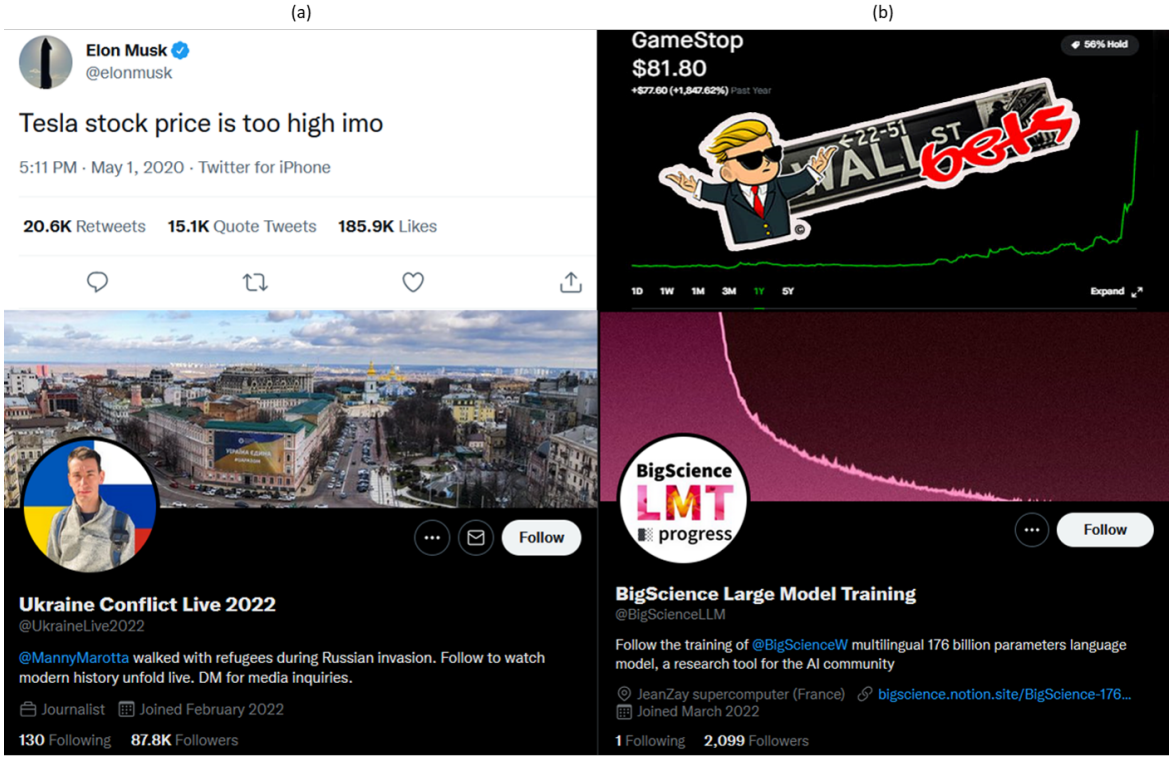

Figure 1.1: (a): Elon Musk decided to tweet that Tesla stock price is too high. It caused a drastic decrease in the valuation of the company and after that, Musk was removed from the presidency of the company. (b): The "Wall Street Bets" event caused a surge in the GameStop stock value. A community of Reddit, a social network, decided to buy shares of GameStop, a company owning several video games shop because they were on the edge of collapsing and several venture capital had bet on that. The surge caused venture capitals to lose millions of dollars. (c): the war in Ukraine attracted a lot of attention, particularly on Twitter where dedicated accounts reported live events. (d): The progress of the training of BigScience Large model was directly reported by the dedicated Twitter account.

type of variation. It must be replicable, applicable to any raw material but also to any component of the supply chain provided that the information about a potential disruption is publicly available online. Once these events are identified, the event detection system must be configurable to focus on the kind of events that can be interesting.

1.2 Challenges

To fulfill these conditions, multiple challenges arise. Concerning the raw materials, a first thing to do is to identify the relevant data sources. Indeed, most of the content discussed online is irrelevant, very few conversations are about events happening in the real-world, and even less content is about an event that could have an impact on the raw materials supply chain. Thus, it is important to identify sources discussing potentially interesting topic, publishing verified data formulated in an informative manner. Some example of interesting sources could be websites that are specialized on economics, business newsfeeds, social media account of traders, news companies or influencial personalities of the domain. To be able to identify interesting sources, we must determine what kind of events will have an impact on raw materials supply chains. This is not a trivial task because multiple types of events can have an impact, depending on the type of raw material we study. For example, the weather will have a huge impact on agricultural commodities while they will probably be less significant for metals or oil.

Once these sources are identified, one must retrieve information from the extracted data. We want to extract information and particularly events in a real-time manner, to allow decision-makers to react as quickly as possible. As we said before, we will focus on textual data streams because they are the privileged media type for quick reactions to events. Due to this characteristic, this type of sources usually contains a lot of noise, the quality of the publication is variable and can change in time. Thus, filtering rules are necessary such as filtering spam and noise, to ensure that the documents analyzed are relevant. It is also necessary to clean the data to ensure that the content is meaningful and can be treated properly, otherwise the quality of the detected event will be poor.

Then, once quality data is extracted from the streams, comes the event detection phase. A first desirable feature of an event detection system is it should be able to find a difference between an event and a mundane conversation. Another challenge is that the events we are looking for are unknown beforehand. Specifying which event to look for can be misleading. The Covid-19 outbreak is a great example of such an event: before that, looking for epidemic outbreak would probably not have been the first concern of raw material buyers. The events can also vary in size, since we are

working with a really specific domains. Thus, we cannot rely only on the popularity of an topic to decide whether it is an event.

When some events are detected, one must ensure that they are tracked in time. An event can be variable in time and thus have evolving consequences. For example, the Suez Canal congestion was first a commercial event, then it moved to an event related to law when the government had to decide of the sanctions for the company running the boat. Thus, it is important to know when an event starts, when it ends, and how it evolves. Once meaningful events are detected, tracked and described, they can be presented to decision-makers to help them in their daily decision process.

1.3 Outline

In this thesis, the major contribution is an event detection system fulfilling the conditions listed above, i.e. detecting events of different sizes in an open-domain manner. This system will be developed in the first chapters of this document. In Chapter 2, we present the related work on event detection, particularly in textual data streams. We identify the major orientations of the existing models, present how they are usually built, they objectives, strengths and limits. In Chapter 3, we present our Event Detection Framework. In this chapter, we also justify the choices we made to build this framework, and introduce the different modules that constitutes it. Once this is done, we present our Event Detection System (EDS) build on top of this framework and focus on each module to highlight the scientific challenges of each of them, and answer these challenges in Chapter 4 and Chapter 5. The code relative to these chapters is accessible online[4]. Then, in Chapter 6, we focus on the challenges linked with the industrial application, namely the study of raw materials. We propose a method to determine important events linked with raw materials, apply this method to specific commodity, the phosphate. Lastly, we introduce, in this chapter, the general process in which the event detection system takes place. Finally, Chapter 7 concludes our work and introduces its possible perspectives.

[4]https://gitlab.com/Emaitre/eventdetectionsystem

Chapter 2

Related work on Event detection in text data stream

In this chapter, we develop the related work about event detection in text data streams. First, we present different definitions of an event. Second, we present the task of event detection. Third, we discuss the usage of social networks as information sources. Then, we present different event detection systems for event detection on social networks. Finally, we present different text representation models.

2.1 Event definition

In the Cambridge dictionary, an **event** is defined as "*anything that happens, especially something important or unusual*". Thus, as we have seen in the introduction section, an event is a moment of particular importance (Champagne, 2000) and depends on the sources discussing it (Nora, 1972). These conceptual definitions are useful to understand the concept of an event. However, in the context of text analysis, one needs a more precise definition to decide whether an event is happening. In the field of Natural Language Processing (NLP), the definition of an event is still an open issue (Sprugnoli and Tonelli, 2017), so several definitions of events exist.

A first common definition is used in the context of "event trigger-based approaches", which is purely a text-based one that supposes that some words, named trigger-words, trigger the event in the sentence and they are carrying the meaning. The associated task is a classification task that consists in classifying words in specified event categories. Detecting and classifying those words hence allows one to understand if a sentence depicts an event. ACE 2005 (Grishman et al., 2005) is the reference dataset for this task. According to the ACE 2005 annotation guideline, in the sentence "A police officer was killed in New Jersey today", an event detection system should be able to recognize the word "killed" as a trigger for the event "Die". This dataset has been used multiple times (Nguyen and Grishman, 2015), (Feng et al., 2016), (Hong et al., 2018), (Kodelja et al., 2019).

Even if this definition is interesting, it does not perfectly fit our context. If we deal with newsfeeds, events are usually described in an informative way and most informative headlines probably contain trigger words. However, working with other data sources such as social networks, people do not necessarily describe the events when they are talking about them. Indeed, as we will see in Section 2.3, restrictions such as the limited number of characters per message force users to post concise messages including little context, or summarize it using keywords such as tags. Thus, in the context of social networks, other definitions of "event" are more common.

In the context of Topic Detection and Tracking (TDT) (Allan, 2012), the event is the root cause for all the discussion topics around it. A topic is defined to be a set of news stories that are strongly related to some seminal real-world event. They give the following example: "When a bomb explodes in a building, that is the seminal real-world event. Any stories that discuss the explosion, or the rescue attempts, the search for perpetrators, or arrests, trials, and so on, are all part of the topic".

Another one was introduced in (Dou et al., 2012): "an occurrence causing a change in the volume of text data that discusses the associated topic at a specific time. This occurrence is characterized by topic and time, and often associated with entities such as people and location". This definition corresponds to the fact that an event is created by the sources talking about it since it considers that a variation in the volume of

conversation dealing with the subject has to happen. One could say that the topic is "trending".

Other definitions exist, such as the one proposed by (McMinn et al., 2013). The definition is divided into two parts: "Definition 1: An event is a significant thing that happens at some specific time and place." "Definition 2: Something is significant if it may be discussed in the media. For example, you may read a news article or watch a news report about it". They identify an event by a group of entities (e.g. people; location) that is discussed in the documents dealing with the event.

As in the previous definitions, the notion of time, place, and significant entities involved in the event are present. However, they do not consider that an event necessarily causes a surge in the discussions about it. The authors of (Fedoryszak et al., 2019) extend this definition in two ways: "first, we argue that a significant thing is happening when a group of people are talking about it in a magnitude that is different from normal levels of conversation about the matter, or in other words, it is trending.", "Second, we claim that the eventful conversation can change over time, and our data model for an event should reflect this."

This extension of the definition gives the definition introduced in (McMinn et al., 2013) the same properties as the one given by (Dou et al., 2012). However, they add the notion that an event can evolve. Indeed, a crisis such as the covid-19 pandemic was first a solely health-related event, and then became an economic-related event as well, and so on. Thus, this extension seems appropriate.

Another definition is given by the authors of (Nolasco and Oliveira, 2019). They define an event as "a significant occurrence limited by time and with an associated location" and introduce the notion of subevent: "A subevent is an event associated with another event by a composition association". Thus, subevents could be seen as all the topics discussing an event, to make a parallel with the definition of topics in TDT.

Considering all these definitions and our context, we decide to keep the definition introduced by (McMinn et al., 2013) and extend it with the second claim introduced by (Fedoryszak et al., 2019). We consider that indeed an event is evolving in time, however, in certain contexts, events might not cause a surge in discussion about the topic, or at least not a surge significant enough to be detected by event detection systems. It is particularly true for domains that are confidential or particularly specialized, such as raw materials, which we are interested in in this thesis. To summarize, we will use the following definition of an event in the rest of this thesis:

Definition 2.1.1. An event is a significant thing that happens at some specific time and place. Something is significant if it may be discussed in the media. For example, you may read a news article or watch a news report about it. It is identified by a group of entities (e.g. people; location) that is discussed in the documents dealing

with the event. The eventful conversation can change over time, and our data model for an event should reflect this.

Now that we have defined what is an event, we focus on the task of event detection, which is a subtask of Topic Detection and Tracking (TDT), presented hereafter.

2.2 The event detection task

2.2.1 Definitions

Event detection (ED) is a task related to Topic Detection and Tracking (TDT). TDT is a body of research and an evaluation paradigm that addresses event-based organization of broadcast news (Allan, 2012). TDT research begins with a constantly arriving stream of text from newswires and from automatic speech-to-text systems that are monitoring television, radio, and Web broadcast news shows. The goal of TDT is to monitor these information sources in order to automatically detect and alert analysts to new and interesting events happening in the world. It is a field that is continuously developing over the years, with the first studies published before 2000 (Allan et al., 1998), (Yang et al., 1998).

TDT is usually divided into five tasks that will help to solve the problem of event-based news organizations:

- **Story segmentation** consists in dividing the transcript of a news show into individual stories.
- **First Story Detection (FSD)** is the problem of recognizing the onset of a new topic in the stream of news stories. The goal of FSD is to recognize when a news topic appears that had not been discussed earlier. A system performing FSD is then evaluated by its ability to detect the first document discussing a news topic and is not interested in what happens in a middle of a topic or its evolution.
- **Cluster Detection** is the problem of grouping all stories as they arrive, based on the topics they discuss. It consists of grouping together all the documents that deal with the same story. When a new story appears, a new group is created for this story. The creation of this group is an unsupervised task, meaning that the number of groups is not known beforehand.
- **Tracking** consists in monitoring the stream of news stories to find additional stories on a topic that was identified using several sample stories.
- **Story Link Detection** is the problem of deciding whether two randomly selected stories discuss the same news topic.

All these tasks are important for event detection, however, we will not consider the task of Story Segmentation in this thesis, due to the context presented at the beginning of this document, i.e. we do not consider transcripts of news shows. All the other tasks will be addressed in this thesis. However, all these aspects are not necessarily treated by all event detection systems and depending on the application, different properties are desirable. To have a better understanding of the characteristics of each type of event detection system, several categorizations exist in the literature. We now present these categorizations.

2.2.2 Categories of event detection systems

In the field of event detection, three categorizations can be distinguished. The first categorization consists in classifying the event detection systems according to the knowledge about the target events. In this context, systems fall between two categories: **Closed Domain Detection** and **Open Domain Detection** (Atefeh and Khreich, 2015). The first ones are mainly focused on detecting specific events such as earthquakes (Sakaki et al., 2010). The second one focuses on detecting events that are not known beforehand (Petrović et al., 2010).

A second categorization depends on the granularity of the data that is used. Systems are again categorized into two categories: **Feature-pivot** (Li et al., 2012) and **Document-pivot** (Petrović et al., 2010). The former focuses on data at the feature level, such as word or sentence segments. These systems aim at detecting features that have an unusual variation in their frequency, which might imply that an event described by this feature is happening. The second focus on the data at the document level, aiming at grouping together documents that are related and then analyzing these groups.

A third way to categorize the systems is according to the temporality of the events to be detected. This time, the categories are **Restrospective Event Detection** (RED) (Becker et al., 2011a) or **New Event Detection** (NED) (Sakaki et al., 2010). RED consists in detecting events that have happened in the past, retrospectively. On the other hand, NED consists in detecting events in real-time.

All of these approaches have their interest depending on the needs of the application. Thus, event detection is a common topic of research in the domain of information retrieval for years now. However, it has gained interest in the last decade notably because of the rise of social networks. Most of the papers published about event detection in the past years focus on event detection on social networks data streams. Social networks are used for event detection because they are really fast to react to breaking news but also because information about short-term and long-term events are discussed on it (Zubiaga et al., 2018). Event detection on social networks is now considered a classic text mining task (Allahyari et al., 2017). It is particularly true

for Twitter due to its structure and its policy. Twitter is indeed considered the most efficient social network for event detection (Hasan et al., 2018).

Considering the context of our application, we will focus on open-domain NED in order to detect new, potentially impacting events in real-time. Since we do not know what will happen, we do not want to specify the type of events we are looking for to ensure that we will not miss any important events. Moreover, because the properties of social networks, presented in the next section, and particularly Twitter are adapted to the application we are interested in, we will develop event detection on social networks with a special emphasis on Twitter. In the next section, we present how social networks are used as information sources and we present in more detail the functioning of Twitter.

2.3 Social networks as information sources for event detection

Several social networks exist and they all have their specificities, guiding how the usage is made of them. Twitter is the most used social network in Information Retrieval Research (Atefeh and Khreich, 2015), (Hasan et al., 2018). Different reasons can explain this, some of them are linked to the structure of Twitter, which will be presented in this section, but also because of Twitter's data policy. Contrary to most other social networks, Twitter offers easy access to a part of its data through its API. Different APIs exist, notably the stream API which allows access to 1% of the current tweets, and the search API, to retrieve tweets from the past using some filtering rules. Even if this access is still restricted, it is better than most of the other social networks. Thus, Twitter is a source of interesting data easily accessible to any researcher.

In this section, we first present why Twitter and other social networks can be considered information sources and why so many people, including professionals such as journalists, use them that way. Then, we describe Twitter and its main features.

2.3.1 Social networks and Twitter as information sources

The usage of social media as information sources has received a growing interest in the past years. The percentage of French adults using social media as an information source is growing since 2013, contrary to most of the other types of sources such as Print or Television, as we can see in Figure 2.1. This may show a growing interest in news covered in a different manner or that are not covered in traditional news media.

This trend is also confirmed for professionals such as journalists: the use of social media sources has resurged massively in recent years (von Nordheim et al., 2018). The authors also add that Twitter is more commonly used as a news source than

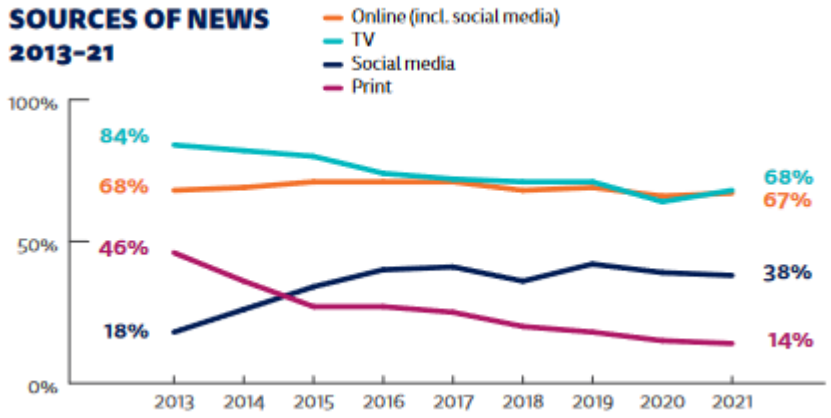

Figure 2.1: Information sources consumption for French people. Source[1]

Facebook, even if Facebook is an overall more popular social network. According to (Castillo et al., 2011), Twitter is an ideal environment for the dissemination of breaking news directly from the news source and/or geographical location of events, due to its characteristics. For some journalists, Twitter has become so normalized that tweets were deemed equally newsworthy as headlines appearing to be from the press agency wire (McGregor and Molyneux, 2020). These practices have positive impacts because Twitter may conduce a wider array of voices into the mainstream news agenda. In the same paper, the authors argue that Twitter also influences journalists' news judgment. Twitter's growing centrality in the news process warrants greater scrutiny from journalists and scholars. According to (Hernández-Fuentes and Monnier, 2020), Twitter usages can be classified into four distinct categories, which correspond to four specific moments of the news production process: the identification of the newsworthy content and relevant sources, the collaborative verification of information, the writing of the article. Twitter is mostly used for news identification, as most journalists have reservations to use it as a source or as a means to identify sources. Most of them consider that the platform has low credibility and ensues a lack of trust from the journalists.

Thus, several reasons justify the use of social media and particularly Twitter as an information source for event detection. Twitter invites users to share news content, due to its structure. The authors of (Kwak et al., 2010) even argue that the structure of Twitter makes it similar to a news media. This can explain why Twitter is the favorite social media of the journalists: it allows them to directly be in contact with the sources but also to build a connection with their audience (Swasy, 2016). A common trend in papers dealing with event detection on Twitter is to consider users as sensors of the information, posting messages when they are activated by important news (Sakaki et al., 2010), (Nolasco and Oliveira, 2019). It is thus very interesting to

[1] `https://reutersinstitute.politics.ox.ac.uk/sites/default/files/2021-06/Digital_News_Report_2021_FINAL.pdf`

have a globalized set of sensors that share in real-time information about events that are happening around them.

Thus, because of all of these properties and the ease of access to the data, Twitter is very popular among researchers. Now, we present the structure of Twitter in more detail.

2.3.2 Twitter

To better understand Twitter[2] no and the elements that make up the social network, we follow the same organization as the one presented in (Edouard, 2017), using some of the elements related there while completing them.

The social network

Twitter is a social network where people communicate using short messages called tweets. It allows public discussions about various topics in tweets that are no longer than 280 characters (140 before 2018). This limitation is intended to promote clever use of the language, focus on precise topics, and easiness of access to the information: according to Twitter's Creative Director Biz Stone, "creativity comes from constraint"[3]. This brevity encourages users to post sometimes multiple tweets a day. According to (Fedoryszak et al., 2019), there are approximately 500 million tweets a day, which correspond to 6000 tweets per second on average. Twitter has a followers/following structure, meaning that anyone who follows an account will see the tweets posted by this account. Relations are asymmetric, contrary to other social networks such as Facebook, thus the Twitter network can be assimilated to a directed social network or follower network (Brzozowski and Romero, 2011). Any user can follow another one without an approval step or a reciprocal connection, meaning that a user will not see the activity of its followers unless they follow them back. On the Twitter landing page, users are invited to post about "what's happening", allowing them to quickly spread information about personal activities, any recent news, event, or comment information shared by other users within their communities (Java et al., 2007).

Twitter has specific features such as retweets, user mentions, hashtags, and hyperlinks, which allows interactions and conversations between users, organization of the content, and pointers to external content. We present these features hereafter.

Retweets

The retweet is the "repost" or "sharing" function of Twitter. Twitter users "retweet" to spread the original tweet to their own community of followers. To retweet a message, a

[2]http://www.twitter.com/

[3]https://www.sfgate.com/living/article/What-is-Biz-Stone-doing-3165704.php

Twitter user can press the retweet button under a tweet to share it, optionally adding a comment to it. In the past, the unofficial convention was to copy and paste the content of the tweet and then add "RT @username" at the beginning of the tweet, along with a possible comment. According to (Boyd et al., 2010), there are multiple reasons to retweet: for the diffusion of the information to different communities or to be part of a conversation; it also allows to spread information to its network, to react to the tweet, to validate the content of the tweet, for self-gain and so on. Retweets can have different utilities, like the measure of the popularity of tweets and users (Kwak et al., 2010). While accessing tweets through the Twitter API, retweets still have the mention "RT @username" and are annotated as retweets in the metadata provided.

User mentions

All tweets are broadcasted to the community of the user posting them and may be public depending on the setting of the account. However, if this user wants to address the tweet to a specific user, a common practice is to mention the account of the user with the user mention "@username". Tweets containing a user mention are considered as a reply or communication directed to the mentioned user (Honey and Herring, 2009). When a user is mentioned in a tweet, it is possible to access its profile using the hyperlink associated to the mention.

Hashtags

Hashtags are probably the most popular element of Twitter. They are used to tag tweets and attach them to the main topics discussed. Hashtags can be of various forms such as single words (#Soccer) or multiple words (#GameOfThrones). Hashtags contain useful information as they are the principal mean used to attach a tweet to a topic. A tweet can contain a various number of hashtags, ranging from zero to several. On the front page of Twitter, trending hashtags are listed to give access to trending topics to users. The hashtag mechanism is now included in most social networks.

Hyperlinks

Tweets are made to react to what is happening, as we can see on the front page of Twitter, and the length constraint limits the context that can be included. Thus, sharing external links is a common practice on Twitter to include additional information in the tweet. URLs can point to different external websites linked with the topic of the tweet, such as newspapers websites or an organization website to promote it. Most of the time, a shortened version of the URL is shared to save characters, using URL shorteners such as tiny.url or bit.ly.

Thus, the structure, the usages, and the data policy of Twitter make it an ideal

playground for journalists and analysts looking for breaking news but also for researchers looking for a fast and automated way to detect events in several domains. However, detecting these events is not a trivial task. Hence, event detection systems are needed to ensure that no event is missed and the globality of the network is analyzed. In the next section, we present different approaches that exist in the literature, with a special emphasis on those adapted to Twitter.

2.4 Event detection on Twitter

Social networks are an important source of information but extracting it from the data stream is not easy. In this section, we present the major trends in research about event detection on social networks. We chose to organize this section in two parts, first **document-pivot event detection systems** and then **feature-pivot event detection systems**. As previously stated, it is a common practice in the literature (Atefeh and Khreich, 2015), (Hasan et al., 2018) to split event detection systems into these two categories, which allows one to highlight the different characteristics considered to group and analyze documents. In particular, we chose this categorization in this thesis because we are interested in doing open-domain, new event detection (NED), which can be performed using both feature-pivot and document-pivot approaches. Presenting the systems according to categorizations in which only one of the two categories can interest us would not make sense. It is important to notice that sometimes, the limit between document-pivot and feature-pivot is thin and this categorization is developed for clarity but is not an end in itself. Indeed, some document-pivot approaches rely on particular features, such as named entities while feature-pivot approaches can keep track of the original document of a feature. The category assigned depends on the aspect on which the emphasis is placed.

Most event detection systems follow a framework similar to the one presented in Figure 2.2. First, tweets are extracted from Twitter's API. Then, they are preprocessed to remove spam or badly formatted documents. Then, event detection technics are applied to separate event documents or event clusters from mundane documents or clusters. Then, these detected events are ranked according to different criteria such as the impact on the crowd, and finally, they are summarized in a meaningful way. Depending on the representation method chosen for the documents, some of these steps are grouped, however, the high-level description of the analysis steps needed stay relevant.

2.4.1 Document-pivot event detection systems

In the document-pivot approach, the most common idea is to produce a representation for each document in a representation space, calculate a distance between the docu-

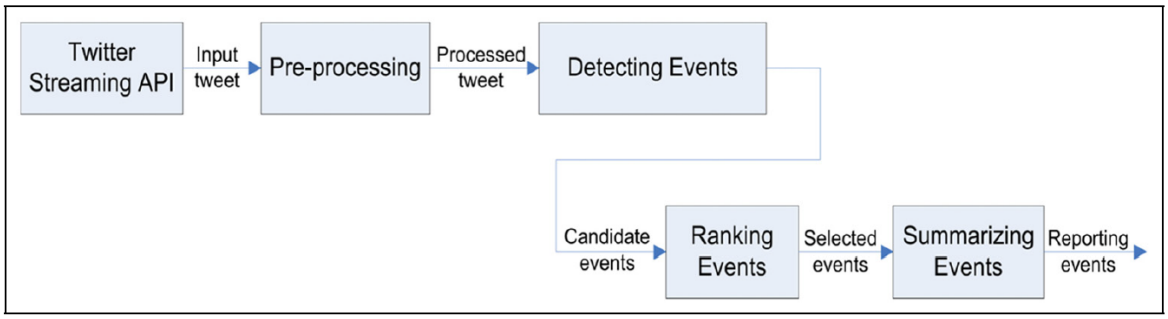

Figure 2.2: A high level representation of Event Detection Frameworks. Extracted from Hasan et al. (2018)

ments, and then group similar documents together using clustering techniques. Once the clusters are created, they are evaluated in order to determine whether they discuss an event. The most common way to represent a document is by using its textual content. We will first present text-centric systems, and then present systems based on other features.

Text-centric systems

One of the first proposed document-pivot approaches is the First Story Detection (FSD) algorithm, which was first introduced by (Allan et al., 2000). The principle is to find the first document discussing an event and then group together new documents discussing the same event. To do so, the problem is modeled as a dynamic clustering task, using nearest neighbors search to group the documents. Then, many authors reused this algorithm, applying it to new domains and speeding it up. (Petrović et al., 2010) apply this algorithm to the task of event detection on Twitter. They improve it using LSH (Locality Sensitive Hashing), which allows a faster nearest neighbor search. To evaluate whether a cluster (which is also called "thread" in the context of FSD) discusses an event, they compute the entropy of the cluster:

$$H_{cluster} = -\sum_i \frac{n_i}{N} log \frac{n_i}{N}, \tag{2.1}$$

where n_i is the number of times word i appears in a cluster, and $N = \sum_i n_i$ is the total number of words in a cluster. They use entropy to rank the clusters and they consider that a cluster with low entropy (< 3.5) is not event-worthy. A low entropy corresponds to a cluster containing very little information, such as a spam cluster. As previously said, the objective of the FSD method is to find the first document referring to an event. However, these methods are usually evaluated not only on their capacity to detect the first document discussing an event, but also on their ability to cluster documents discussing the same event. Thus, it is actually a combination of the FSD task and the Clustering task described in Section 2.2. In (Repp and Ramampiaro, 2018), the authors propose to speed up the FSD algorithm using the

mini-batch approach. This version of the algorithm is presented in Algorithm 1. In (Hasan et al., 2019), the authors use the FSD algorithm to evaluate the novelty of a tweet and then assign the tweet to a cluster depending on the difference between the representation of the tweet and the representation of the center of the cluster. They use a three steps filter to evaluate whether a cluster discusses an event. They use the entropy measure and the user diversity defined as follows in (Kumar et al., 2014):

$$U_{cluster} = -\sum_{u} \frac{n_u}{N_t} log \frac{n_u}{N_t}, \tag{2.2}$$

where u is a user who posted a tweet in the cluster, n_u is the number of tweets published by the user u which are part of the cluster, and N_t is the total number of tweets in the cluster. They discard clusters with low user diversity (≤ 0). A positive user diversity value ensures that a cluster contains tweets from more than one user. They also use a combination of features such as the number of tweets in a cluster, the presence of URLs directed to a news portal, and the time span between the first and last tweet of the cluster. They also take into account the Longest Common Subsequence in the tweet as they noticed that news reports tend to follow the same structure.

In all these papers, the tweet content is represented using TF-IDF, a classical text representation in Information Retrieval (Baeza-Yates and Ribeiro-Neto, 1999), which is a weighted bag of words. In (Mazoyer et al., 2020b), the authors also use the mini-batch version of the algorithm and compare the performances of different text representation models. They compare the performances of Word2vec, TF-IDF, ELMO, BERT, S-BERT, and Universal Sentence encoder. The general assumption of these models is that they represent semantic information contained in the text contrary to TF-IDF which is lexical. These models are presented in the next section. They also produce dense vectors while TF-IDF produces sparse vectors. The authors conclude that representation models based on recent architectures such as Transformers perform poorer than TF-IDF in the context of FSD. Contrary to the previous approaches, they do not perform an event detection step. They work on a dataset where a gold standard of tweets is annotated as event-related, thus they consider that the documents inside the clusters are necessarily discussing an event.

We will present different text representation methods in detail in section 2.5, including the one used in this paper.

The FSD algorithm is very common and used in the literature but other approaches exist. The authors of (Becker et al., 2011b) use TF-IDF as well to represent the content of the tweets and then cluster topically similar tweets together. To do so, they use an online incremental clustering algorithm. Because the number of clusters is not known beforehand, they chose an algorithm based on an empirically calculated threshold value to find a suitable cluster assignment for each tweet, based on its similarity with other tweets of the cluster. Then, they classify these clusters using Support Vector Machine

Algorithm 1: First Story Detection, (Repp and Ramampiaro, 2018)

input : threshold t, window size w, corpus C of documents in chronological order
output: thread ids for each document

$T \longleftarrow []$;
$i \longleftarrow 0$;
while *document d in C* **do**
  if *T is empty* **then**
    $thread\_id(d) \longleftarrow i$;
    $i \longleftarrow i + 1$;
  else
    $d_{nearest} \longleftarrow$ nearest neighbor of d in T ;
    if $cosine(d, d_{nearest} < t)$ **then**
      $thread\_id(d) \longleftarrow thread\_id(d_{nearest})$;
    else
      $thread\_id(d) \longleftarrow i$;
      $i \longleftarrow i + 1$;
    end
  end
  if $|T| \geq w$ **then**
    remove first document from T;
  end
  add d to T;
end

(SVM) trained on an annotated dataset. In (McMinn and Jose, 2015), the authors also represent tweets using TF-IDF and named entities (NE) then use an incremental clustering algorithm, based on similarity criteria but also the length of the tweets. To be able to process the tweet in a real-time manner, they filter out tweets that do not contain any NE. They consider that a tweet does not carry important information for the identification of events if they do not contain any named entity, following their definition of an event introduced in (McMinn et al., 2013) that we presented in Section 2.1. In (Boom et al., 2016), the authors introduce a method based on semantic word embeddings and frequency information to arrive at low-dimensional representations for short texts designed to capture semantic similarity. They learn a representation for the words in the documents and then weigh them based on their TF-IDF score. In terms of datasets, they only consider tweets published by English news agencies, so they do not perform an event detection step. They consider that two tweets are semantically related if they are generated by the same event. To the best of our knowledge, it is the first attempt to combinate TF-IDF and semantic representation of words in the context of short text messages. In another type of approach, the authors of (Zhou et al., 2017) give a structured representation of events. They extract events from Twitter using a non-parametric Bayesian Mixture Model with Word Embeddings. They create event clusters from tweets and the events are modeled as a 4-tuple $< y, l, k, d >$, respectively modeling non-location NEs, location NEs, event keywords, and date. The components of the quadruple are generated using a multinomial distribution computed using a Dirichlet process. There is no event detection step in this paper since they only work with event-related tweets. Following the same idea of representing events using structured representation, the authors of (Li et al., 2017) include semantics by splitting tweets terms reflecting one or more event aspects. The semantic classes include named entities, mentions, locations, hashtags, verbs, nouns, and embedded links. They group tweets into clusters using class-wise similarity. Then, they filter old stories using a temporal identification module. A novelty score is computed at the cluster level to determine new events.

Thus, a lot of work mostly consider text-based similarity to group document that discuss the same topics and events together. However, other approaches exist exploiting different features.

Methods exploiting other features

In (Becker et al., 2010), the authors propose to learn similarity metrics for the clustering of event-related tweets. They combine textual features, temporal features, and location features of the documents. They consider that documents dealing with the same event should have a similar publication date and should be published from a similar location. In (Cai et al., 2015), the authors argue that apart from the textual

content, tweets contain other features such as image, timestamp, location, and hashtag and that incorporating different features is useful and helps to have a better understanding of events. They also argue that no model has comprehensively exploited all these features in one framework. They introduce a novel topic model which jointly models text, image, location, timestamp, and hashtag to discover events from the sheer amount of tweets. In practice, they first use spatio-temporal multimodal Twitter LDA for event detection, based on the features presented earlier. Then, they track the events using a maximum-weighted bipartite graph matching and finally visualize events using representative images. In (Guille and Favre, 2014), the authors perform event detection, tracking, and visualization on Twitter. Their event detection system, named MABED, not only uses the textual content of the tweets but also the social aspect: MABED relies on dynamic links (i.e. user mentions) and does not presuppose a predetermined duration for the events. Events are calculated using a statistical measure to find anomalies in mentions between users. MABED describes each event by one or more main words and a set of weighted related words, a period of time, and the magnitude of its impact on the crowd. To ensure an efficient exploration of the detected events, they propose three interactive visualizations: a timeline that allows exploring events through time, a chart that plots the magnitude of the impact of events through time, and a graph that allows identifying semantically related events. In more recent work (Han et al., 2019), the authors work on geotagged tweets. They show that when an event is happening, the time series describing the number of tweets posted in the area of the event follows a power law. It also works when dividing the region of the publication, allowing one to determine with precision where the event is happening.

Thus, even if some works propose approaches based on other features rather than text to group tweets together to detect events, the vast majority of the work based on document-pivot approaches include textual similarity to group tweets together, making it a critical aspect to build an efficient event detection system.

We will now present feature-pivot approaches, to have a good overview of all the approaches that exist in the literature.

2.4.2 Feature-pivot

Contrary to document-pivot approaches, feature-pivot approaches focus on the analysis of features within the documents and grouping them according to their distribution. Most of the features used in related work are also based on textual content, so we divide this section just like the previous one.

Text-centric methods

The most classical feature-pivot approach is called Twevent (Li et al., 2012). In this paper, the authors introduce segmentation-based event detection from tweets method. Tweets are segmented using a "stickiness" score of segments and bursty segments are selected based on the prior probability distribution of segments and user diversity and are clustered into events. Newsworthy events are then computed using two newsworthiness scores based on the segments composing it: one for the segment and one for the event. The authors of (Morabia et al., 2019) enrich this approach using Wikipedia to analyze segments present in articles' titles to tackle problems such as noisy data, informal writing, grammatical errors, and lack of context. (Pandya et al., 2020) use information extracted from external sources such as DBPedia and Wordnet for similar motivations.

Another classical approach has been introduced by the authors of (Weng and Lee, 2011). They propose an event detection based on clusters of discrete wavelet signals build from individual words contained in the tweets. Wavelet transformations are localized in both time and frequency domains, hence allow to identify the time and the duration of a bursty event within the signal. Trivial words are filtered based on signals cross-correlation. The remaining words are then clustered to form events with a modularity-based graph partitioning technique, splitting the graph into subgraphs to create a graph per event.

In (Edouard et al., 2017), the authors extract named entities from the text of the tweets and create a graph using these named entities as nodes with their k-nearest neighbors words as context. Then, they draw an edge between nodes if there are some co-occurrences between the words composing the nodes to obtain a temporal event graph. Finally, they process this event graph to detect clusters of tweets describing the same events. In (Saeed et al., 2019a) the authors also use a graph-based event detection approach. The objective is to detect significant changes in the streaming of tweets. The particularity of their approach is once an event is detected, they delete it to let place for new events. They use a sliding window over the stream of documents to make a temporal aggregating and construct a graph between keywords. They compare the graph of a time window to the graph from the previous one to detect the variations. Another graph-based approach is developed in (Asgari-Chenaghlu et al., 2020). The authors build a memory graph in which nodes represent words and vertices the co-occurrence relation of these nodes. Words meaning is represented using the sum of the BERT embeddings of the documents in which they appear. As words appear in the stream, a larger graph is built. Then, a community detection algorithm is performed to create clusters of entities. Entities are labeled using multimodal data such as images and text. They do not perform the event detection step as they focus on topic detection and not event detection.

In (Fedoryszak et al., 2019), the authors introduce another approach in which they propose to analyze important words in each tweet based on trending words and calculate similarities based on co-occurrences of these terms. They use a community algorithm to cluster these words. They sample the incoming stream using time windows. Thus, to follow the evolution of the events, they create chains of clusters. To link the clusters together, they use the entities composing each cluster, similar to their clustering step. In (Nolasco and Oliveira, 2019), the authors introduce an approach to detect subevents. They argue that most of the event detection approaches focus on main event detection while detecting subevents provides a lot of information about the unfolding of the events. They introduce a subevent detection system based on topic modeling techniques such as LDA. They use this to label each subevents for a better understanding. They argue that their methods allow better tracking of the evolution of the events. The authors of (Kuang et al., 2020) detect bursty features on social networks and then group them into bursty events. Then, the veracity of these events is checked using data from newsfeeds. Data coming from both channels are aligned using supervised learning techniques at the event level. As we have seen previously in this section, there is a certain lack of trust in information shared on Twitter. Thus, using a model to combine news and event detection is interesting to validate the detected events. In recent work (Hettiarachchi et al., 2021), the authors integrate semantic information in the study of events on social networks. They use word embeddings and hierarchical agglomerative clustering to create clusters of documents. Then, they use a time-based sliding window model and take into account temporal variation of the clusters and vocabulary changes to identify events.

Thus, a lot of the feature-pivot approaches mostly use text-based approaches. We now present approaches based on other features.

Methods exploiting other features

In their paper (Chen and Neill, 2014), the authors model Twitter as a heterogeneous graph to detect events. Each node is a feature, such as a hashtag, a user, or the text content of a tweet. They calculate a normality distribution for each element using a historic they calculated on training data. If the neighborhood of a node is too different from the usual distribution, a signal triggers an abnormality warning meaning that an event is happening. In the paper (Shao et al., 2017), the authors argue that naturally, social media are structured as dynamic multivariate networks with: 1) vertices, such as users or locations; 2) relationships, such as spatial neighborhood and followers; 3) attributes, such as frequencies of domain-specific keywords, which evolve over time. They also argue that based on the dynamic multivariate networks, events can be represented as evolving anomalous subgraphs (e.g., connected subsets of vertices with abnormally high frequencies of domain-specific keywords), and they formulate the

problem of event detection and forecasting as the detection of the most anomalous evolving subgraphs in dynamic multi-variate social media networks.

2.4.3 Synthesis

Both approaches are viable as we can see from the density of the work presented. A sample of the results of each section is proposed respectively in Table 2.1 for the document-pivot approaches and in Table 2.2 for feature-pivot approaches. The purpose of these tables is to show the interdependence between some steps of the process. The first important point is the type of data: in most of the approaches, data are both event or non-event-related. This is presented in the first column. In this case, it is necessary to filter the data to reduce spam and noise. This is presented in the second column. The second important step is the discretization of the stream. This step is directly dependent on the clustering technique employed. A sliding time window is used when the event detection task is modeled as a dynamic clustering task. In these cases, the documents are treated incrementally as they arrive. In the case of classical time windows, they are usually disjoint from each other. The event detection task is then modeled as a classical event detection problem. This is presented in the third column of the tables. Finally, the features represented can also have an influence. When working with feature-pivot approaches, the focus is usually made on bursty features thus the filtering of the event/non-event cluster is included in this step. When working with document-pivot approaches, another step of event detection is needed. Hence, for both approaches, a filtering step is necessary but it is not necessarily dissociated from the clustering phase. In any case, the textual content of the documents is the major feature used to group and analyze documents and clusters.

Thus, even if the overall steps are the same for each method, the choices made for each of them are interdependent. Hence, it is important to take into account the whole event detection process when presenting each step. The problems raised will depend on these choices. In the next chapters, we will consider the version of the FSD algorithm introduced by (Repp and Ramampiaro, 2018) presented in Algorithm 1, and particularly the implementation proposed in (Mazoyer et al., 2020b) as our baseline. It is indeed a very classical algorithm and it is used multiple times. We will compare it to the event detection system we introduce in the next chapters.

The next part will be devoted to the presentation of classical text representation models. Then, we will conclude this section and introduce the next section in which we will present the architecture of our event detection system.

Table 2.1: Comparison of different document-pivot event detection approaches from the literature

Article	Type of Data	Preprocessing steps	Discretisation of the stream	Clustering features	Event detection step
Petrović et al. (2010)	Event and non event related	Data cleaning	Sliding time window	Text	Ranking
Hasan et al. (2019)	Event and non event related	Data filtering & cleaning	Sliding time window	Text	Filtering & Ranking
Mazoyer et al. (2020b)	Event related	Data cleaning	Sliding time window	Text	-
McMinn and Jose (2015)	Event and non event related	Data filtering & cleaning	Time window	Text	Filtering
Naaman et al. (2011)	Local event and non event related	Data filtering & cleaning	Time window	Text, Twitter Specific	Filtering
Boom et al. (2016)	Event related	Data filtering & cleaning	Sliding time window	Text	-
Zhou et al. (2017)	Event related	Data filtering & cleaning	Time window	Text	-
Li et al. (2017)	Event and non event related	Data filtering & cleaning	Sliding time window	Text	Filtering & Ranking
Becker et al. (2010)	Event related	-	Sliding time window	Text, Time, Location	-
Guille and Favre (2014)	Event and non event related	Data filtering & cleaning	Time window	Text, Twitter specific	Filtering
Cai et al. (2015)	Event and non event related	Data filtering & cleaning	Sliding time window	Text, Image, Time, Location, Hashtags	Filtering
Han et al. (2019)	Event and non event related	Data filtering & cleaning	Time window	Location	Filtering

Table 2.2: Comparison of different feature-pivot event detection approaches from the literature

Article	Type of Data	Preprocessing steps	Discretisation of the stream	Clustering features	Event detection step
Li et al. (2012)	Local event & non event related	Included in Event detection	Time window & sub time window	Text	Filtering
Morabia et al. (2019)	Event & non event related	Data cleaning	Time window & sub time window	Text	Filtering
Pandya et al. (2020)	Event & non event related	Data filtering & cleaning	Time window & sub time window	Text	Filtering
Weng and Lee (2011)	Event & non event related	Data filtering & cleaning	Time window	Text	Filtering
Edouard et al. (2017)	Event & non event related	Data cleaning	Time window	Text	Filtering & Ranking
Saeed et al. (2019a)	Event & non event related	Data cleaning & filtering	Sliding time window	Text	Filtering
Asgari-Chenaghlu et al. (2020)	Event & non event related	Data filtering & cleaning	Sliding time window	Text	-
Fedoryszak et al. (2019)	Event & non event related	Data filtering & cleaning	Time window	Text	Filtering & Ranking
Nolasco and Oliveira (2019)	Event & non event related	Data filtering & cleaning	Time window	Text	Filtering & Ranking
Kuang et al. (2020)	Event & non event related	Data filtering & cleaning	Time window	Text	Filtering
Hettiarachchi et al. (2021)	Event & non event related	Data filtering & cleaning	Time window	Text	Filtering

2.5 Text representation models

How to meaningfully represent the text content of a document is one of the major issues in information retrieval. The current reference method is TF-IDF (Jones, 1972) which is an improvement of the Bag Of Words (Harris, 1954). TF-IDF takes into account the importance of the words in the representation of the document by weighting each word in inverse proportion to the number of documents in which the words appear. Thus, a word appearing frequently in a document while it appears only a little in the corpus is considered as carrying a lot of information about this document. This word will be highly weighted in the TF-IDF representation of the document. TF-IDF vectors are sparse in the context of Twitter due to the large vocabulary and short size of the documents. This representation is widely used, even nowadays, in information retrieval and obtains very good performances, particularly on short texts extracted from social networks.

These statistical representations are currently complemented by dense vector representations, called word embeddings, based on deep learning approaches. The authors of (Mikolov et al., 2013) introduce the Word2vec model which corresponds to a neural approach allowing to associate a word with a vector, which is computed depending on the context in which the word appears in the training set. Thus, the vector representing a word contains information about it. The assumption made for the constitution of these vectors is that words whose contextual use is close will carry similar meanings and thus will be represented by a close vector. Variations exist, such as the FastText (Bojanowski et al., 2016) model which splits words into n-grams, allowing to take into account the construction of words, especially suffixes and prefixes.

The most recent models based on neural networks are based on Transformer architecture (Vaswani et al., 2017). This architecture is currently replacing neural architecture such as Recurrent Neural Networks (RNN) in the context of Natural Language Processing and more recently also replacing Convolutional Neural Networks (CNN) in the context of images. Overall, Transformers are becoming the go-to architecture for most of the new neural network architectures. The Transformers has multiple interests according to (Vaswani et al., 2017):

- They are better for sequential operations than RNN since the complexity of the Transformers is better when the representation dimension is superior to the length of the sequence. It is the case most of the time in NLP since the representation dimension is a few hundred. A sequence of words is rarely so long. Moreover, they can be easily parallelized as they contain only matrix multiplications and no sequential operations.

- At best, the convolution layers have the same complexity as self-attention layers

that are the building blocks of the Transformers. When the sequence size is superior to the size of the convolution kernel, the self-attention layers have a better complexity.

- Each "head" (which constitutes the self-attention layer) of the Transformer has a particular task. Since each head is based on the attention mechanism (which can be seen as a weight matrix), it is possible to interpret to what contributes each head.

The most notable implementation of the Transformers in NLP is BERT (Devlin et al., 2018). BERT is a language model based on the principle of Transfer Learning (Pan and Yang, 2010). The idea is that learning some general tasks and then applying this knowledge to a more specific task can improve the performances of the model on the downstream task. The principle is presented in Figure 2.3. The major interest of BERT is that it is a pre-trained model that can be applied to different NLP tasks. Indeed, the model is first pre-trained on two types of tasks, predicting the hidden words in a sentence and predicting the next sentence given the previous one. BERT was pre-trained on two datasets, BookCorpus containing 800 million words, and the whole English Wikipedia, containing 2500 million words. Then, it is possible to fine-tune the model on a specific task. Since the publication of BERT, several papers worked on this model and showed that training longer larger models with more data improved the performances (Liu et al., 2019). This model is named RoBERTa. Some other work showed that the knowledge contained in such a model could be distilled to reduce the size of the model (Jiao et al., 2020, Sanh et al., 2019) while conserving most of the performances. Some work proposed models based on the BERT architecture for English tweets. The first model is BERTweet (Nguyen et al., 2020). It is pre-trained on 850M English Tweets, it uses the BERT model configuration, and is pre-trained using the RoBERTa procedure. This model is tested on the tasks of Part-of-speech tagging, Named-entity recognition, and text classification. Another similar model is tweetBERT (Qudar and Mago, 2020). It was trained on several datasets, including datasets from Twitter, but also scientific and biomedical datasets.

Most of the presented models allow to represent words but do not necessarily allow to represent sentences, which could be interesting in the context of short text documents such as tweets. One of the first approaches which focused on sentence representation is Skip-Thought, proposed by (Kiros et al., 2015). It is an encoder-decoder architecture, trained in an unsupervised way to predict the neighboring sentences of a given sentence in a text. Another classical approach is the use of Siamese networks (Bromley et al., 1994), i.e. two neural networks in parallel,

[4]https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a

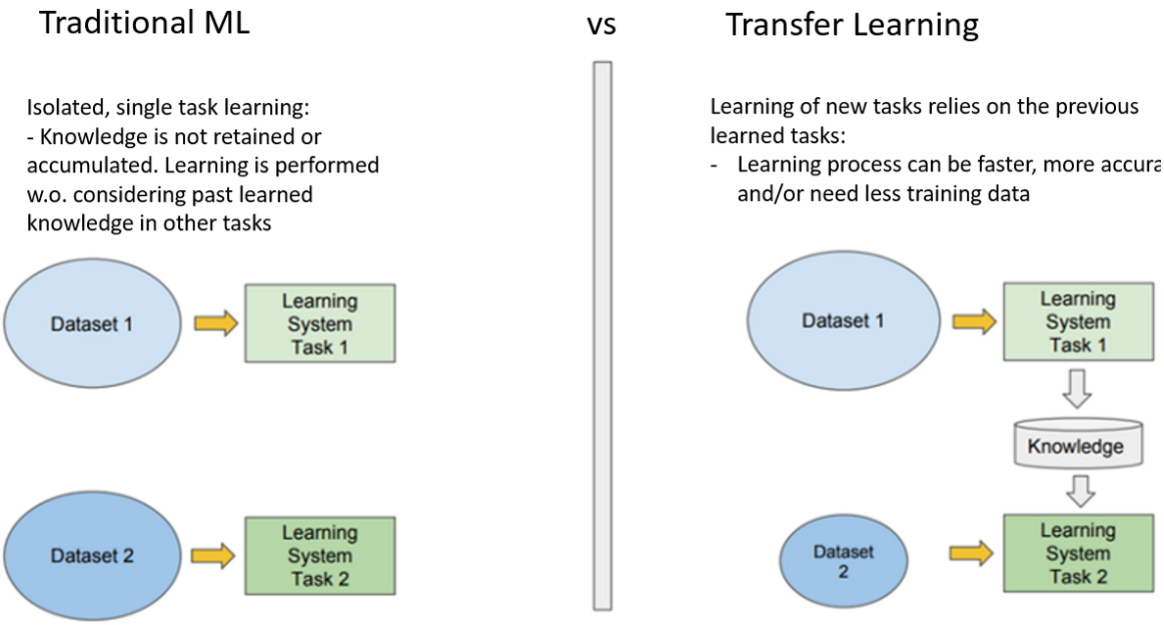

Figure 2.3: Traditional Learning vs Transfer Learning. Source[4]

having the same architecture and the same weights, but which will not take the same input. This is notably what has been proposed by (Conneau et al., 2017) with their InferSent model. It is a two-way Siamese LSTM (Hochreiter and Schmidhuber, 1997) network (a classical RNN architecture) trained in a supervised manner on the SNLI dataset (Bowman et al., 2015). This dataset contains 570,000 pairs of sentences annotated according to three categories: implication, contradiction, or neutral relationship between the two sentences. Another way to represent sentences is to use an architecture based on Transformers. Universal Sentence Encoder (USE) (Cer et al., 2018) is trained on two types of tasks, a supervised one, based on the SNLI dataset in the same way as Infersent, and on unsupervised tasks, like Skip-Thought, which notably include social network documents. Transformers architectures can also be used in the form of Siamese networks. The vanilla BERT architecture performs poorly on short documents of the size similar to a sentence and performs better with longer documents so another approach is needed. More recently, the authors of (Reimers and Gurevych, 2019) propose S-BERT (Sentence BERT) which consists in creating a Siamese network of two BERT models that will be trained with the objective of producing similar vectors for sentences whose meaning is close and dissimilar vectors for sentences whose meaning is distant. Then, a last layer of neurons is added, so that it can be refined on specific tasks. It is worth noting that, to the best of our knowledge, no model based on the Transformers and pre-trained on Twitter datasets is designed for the production of embeddings of tweets. This is probably due to the absence of a dataset similar to SNLI but constituted of tweets.

Thus, there is a great diversity of text representation models and a lot of them

could be interesting for our work, particularly Transformers for the representation of sentences or tweets.

2.6 Conclusion

In this section, we reviewed several works from the literature. First, we discussed the different definitions of event that exists and decided to use definition 2.1.1, proposed in McMinn et al. (2013) and completed in (Fedoryszak et al., 2019): **"An event is a significant thing that happens at some specific time and place. Something is significant if it may be discussed in the media. For example, you may read a news article or watch a news report about it. The eventful conversation can change over time, and our data model for an event should reflect this".**

Then, we showed that the event detection task on text stream attracts a lot of research, with a current focus on social networks, especially Twitter. We showed why Twitter is a privileged information source and reviewed different event detection approaches.

As we demonstrated in Section 2.4, event detection approaches follow the same overall framework but the choices made for each module have a direct impact on all other modules. Thus, the next chapter will be dedicated to the presentation of the event detection framework we follow. This presentation is needed before introducing the scientific problems investigated in this thesis. In this chapter, we justify the different choices we made and discuss which problems we will solve.

We follow this overall way of thinking:

- The current trend in NLP is to consider as much information as possible when analyzing a document. This trend is currently due to the Transformers architectures, which allow getting rid of the constraints linked to recurrent networks. Thus, we want to propose a document-pivot approach that suits better this trend.
- The only document-pivot approach which tried to use Transformer-based architecture in the context of event detection on social media concluded that they are less efficient than classical methods such as TF-IDF (Mazoyer et al., 2020b). They conducted these experiments in the context of a dynamic clustering approach. We want to experiment whether we can find other another approach that performs better.
- One of the principal strength of Transformer-based language models is that they can be fine-tuned to a specific task. We want to take advantage of this asset.
- Event detection approaches need annotated datasets to be evaluated. Due to the amount of data posted on social networks, it is challenging to evaluate correctly event detection systems as it is impossible to label manually all the data.

We propose a method that allows the evaluation of event detection systems on partially annotated datasets.

Now we present the event detection framework we follow.

Chapter 3

A Framework for Event Detection Systems

The objective of this chapter is to present the framework on top of which our event detection system is built. First, we present a general description of the framework. Second, we develop the goals and the problems raised by each component of the framework.

3.1 General framework for event detection

We introduced several constraints induced by our context and summarize them hereafter: first, the event detection system must be able to detect events without specifying what type of event to look for; this is a task named **open-domain event detection**. We decided to perform open domain event detection because in most cases, one does not know beforehand what kind and what number of events will be important or will be impacting in their context. Moreover, since some events never happened in the past so it is complicated to anticipate them. The second aspect is that we want our system to be able to detect both large and small events. Thus, the volume of the text produced by the event can be different and must not be the only criteria to determine whether a discussion topic is an event or not. Finally, we want our system to detect both events that are general, such as the covid-19 pandemic or the congestion of the Suez Canal, but also very specific events that are related to specific domains such as the failure of an important supplier in a defined context.

To adapt to these constraints, we introduce an event detection framework described in Figure 3.1. It is a high-level representation of the framework composed of different modules, in which every module has its own utility and can be realized using different methods, as we have seen in Section 2.4.

We chose to follow a document pivot approach. While feature pivot approaches are also viable, as we have seen in Section 2.4, we think that document pivot approaches

have certain advantages such as being able to consider the whole document in the analysis, allowing us to easily take into account the full content of the document, including more features such as the metadata and embed more context in the representation. It is also less reliant on the burst of some features, and is thus better adapted to detect events of different sizes, that does not necessarily provoke a surge of conversation about the event topic. One of the major drawbacks of document pivot approaches compared to feature pivot approaches is that these methods require to tune a lot of parameters (Fedoryszak et al., 2019). We will be particularly careful about this assumption while designing our event detection method.

Hence, the framework is composed of different modules, presented in Figure 3.1:

- **Document sources** - The goal of this step is to extract documents from different sources. It can be Twitter's API, or a newsfeed such as Reuters.
- **Filtering and preprocessing** - A filtering step to filter spams and uninteresting documents when necessary and a preprocessing step to clean the content of the input documents.
- **Document representation** - The goal is to represent documents in a meaningful way, allowing them to be clustered. Different features can be used. It is a critical step because the quality of the clustering directly depends on it.
- **Document clustering** - In this step, clusters of documents are created. Documents dealing with a similar event should be in the same cluster. The major challenge of this step is to cluster documents without knowing the number of target clusters.
- **Event detection** - Once the clusters of documents are created, they are analyzed to determine whether it deals with an event.
- **Event summarization and tracking** - The goal of this module is twofold: a step to decide whether an event cluster deals with an event that has already been detected, to group them if necessary. Then, the summarization step is used to represent the event in a way that a human can understand easily.

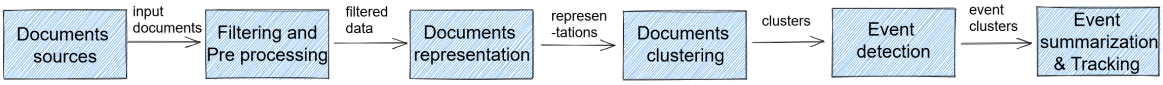

Figure 3.1: A high-level representation of the event detection framework.

All of these modules can have different implementations, however, the choices made for some modules have a direct influence on the possibilities for the others. For example, the type of document representation chosen has a direct influence on the choice

of the clustering algorithm. Thus, instead of studying each module independently, we decide to divide this framework in different phases, which will be studied in the rest of this thesis. We propose to divide this framework into 3 phases:

- Phase 1: Data retrieval, extraction and filtering
- Phase 2: Documents representation & clustering
- Phase 3: Event identification

Figure 3.2 presents the correspondences between these phases and the different modules of the framework.

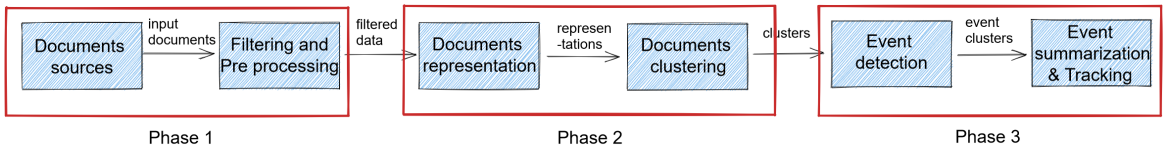

Figure 3.2: The different phases of the Event Detection Framework.

In the rest of this chapter, we develop each of these phases to highlight their content but also the problems that are inherent to them. Then, we present the problems addresses in this manuscript and the ones we let for future work. Finally, we conclude and introduce our work to address those problems in following chapters.

3.2 Phase 1 - Data retrieval, extraction and filtering

3.2.1 Description

The major objective of this phase is to be able to send quality and relevant documents to the rest of the event detection framework. To do so, several steps are needed.

First, we have to determine which sources are interesting and contain the information we want to monitor. These sources can be multiple, from newsfeeds to social networks. It is important to filter these sources for several reasons. When we work with specialized domains such as raw materials, not all the sources will discuss topics that can be interesting to this domain. Moreover, some sources might spread false information, particularly when working with social networks. Several studies showed that some users are more important than others in terms of influence and ability to spread information across the network (Jürgens et al., 2011), (Guille et al., 2013). One can also argue that Twitter accounts of news companies are a reliable source of information. Thus, sources must be chosen carefully, both in terms of media type (newsfeeds, social networks) and also in terms of entities supervised (accounts, newspapers ...).

Second, according to the type of sources, additional filtering might be needed to filter uninteresting documents. When working with newsfeeds, the filters are quite

easy to determine considering that most of the articles are discussing important and interesting topics. When working with social networks and particularly with Twitter, new challenges arise. The first is that social networks are full of spam and most of their content does not contain any interesting information. Hence, incorporating a spam-filtering step is necessary for any event detection pipeline. A second challenge is that according to (Liu and Lapata, 2019), the proportion of tweets concerning news is less than 0.2%. As described in the previous section, we are not only interested in tweets discussing what is in the media but also in tweets that discuss a potentially newsworthy event, so the proportion of tweets we are interested in is probably higher than 0.2%. However, it is still necessary to filter, to filter out mundane conversations and focus on newsworthy documents.

Finally, when working with representation models, some parts of the document does not convey any information and can be considered as noise which has to be filtered to improve the performances. As an example, for models working at the word levels, a classical cleaning step is to remove words known as "stop-words", that are not carrying important information (such as "the", "it" ...). When working with sentence-level language models, they are used to work with well-structured sentences, usually found in the literature. They are not used to work on the syntax that can be used online, which might include abbreviations, mentions to users (@username on Twitter), tags (#hashtags) ... To ensure the quality of the representation, preprocessing steps are needed.

Thus, at the end of this phase, the outputs obtained are documents that are extracted from potentially interesting sources, the text content, and the other features constituting these documents, presented in an exploitable way for the processing steps of phase 2.

3.2.2 Problems raised concerning data retrieval, extraction and filtering

The specificity of our research topic raises the following questions:

- Which sources are relevant to monitor events affecting raw materials stocks?
- Is it possible to retrieve data that allow both detecting specific events (e.g. specific to one special commodity) and general event (impacting the stock market as a whole)?
- How to clean and filter the data to ensure its quality for event detection?

3.2.3 Solutions considered concerning data retrieval, extraction and filtering

To determine which sources to supervise to detect events that might have an impact on supply chains and raw materials stock, the first step is to determine which events are actually impacting. To answer this question, we proceeded in the following way. First, we interviewed several buyers from Scalian, to better understand their daily work, their needs and have a grasp of their expertise, which will give the first orientation of this research.

Then, using the outcome and the lessons learned from these interviews, we decided to make a historical analysis of the events that influenced the raw materials stock market in the past. We also decided to focus on a specific raw material, phosphate. It has several properties that we find particularly interesting.

We address these problems in Chapter 6 of the thesis. The problems linked with data cleaning and filtering will be addressed both in Chapter 4 and Chapter 3. Now, we develop the content of Phase 2 and Phase 3.

3.3 Phase 2 - Data representation and clustering

3.3.1 Description

The objective of this phase is to group documents that deal with the same topic. It is a crucial step because most event detection approaches include a clustering step.

The first step to consider is document representation, in which the task is to extract a meaningful representation of the document from its content. It is a critical step because the clustering performances are directly affected by the quality of these representations. As we have seen in Section 2.4, different approaches can be considered as several aspects can be exploited to extract the content of the documents, such as text, metadata, specific features (URLs, hashtags), or images. The most classical challenge of this task is related to the text representation because most of the event detection methods focus on text features. As NLP models are currently rapidly evolving, there is no consensus on which text representation is the more adapted to this task. A second challenge offered by this task is to be able to jointly exploit different aspects of the document. It is particularly true in the context of social networks such as Twitter where documents are short, which leads users to include little context. It is necessary to find a unified way to represent the different aspects of the documents that allows a good grouping of similar documents.

The second step is to group similar documents. An important challenge of this step is that in the task of open-domain detection, the number of clusters is not known beforehand. Thus, it is necessary to find ways to group documents without specifying

the target number of clusters.

At the end of this phase, clusters of similar documents are created. Then, in the next phase, they will be analyzed to determine whether they deal with an event.

3.3.2 Problems raised concerning documents representation & clustering

The major issue of this phase is "how to represent documents in a meaningful way". As we saw in Section 2.5, several text representation models exist and a lot of work tried to use them in an event detection context. The first result we want to explore is the one obtained by (Mazoyer et al., 2020b). Their results show that Transformer-based architectures are poor performers in the context of First Story detection. As Transformers are currently obtaining state-of-the-art results in most of the NLP tasks, this result can be surprising. Our first motivation is to explore whether we can find a configuration in which Transformer-based language models perform well, which would probably improve the overall event detection.

The second aspect we want to address is whether is it interesting to fine-tune a language model in the context of event detection on social networks. Indeed, the target events evolve over time, a phenomenon usually called "concept drift". In this context, training data and test data are usually very different.

The last aspect we want to address is whether combining different representation models is beneficial to the clustering task. Indeed, as we saw in the literature, several text representations exist and they are supposed to encode different aspects of the text (Lexical, semantic ...). We also want to include in our study specific features, such as tags or URLs that are carrying information.

3.3.3 Solutions considered concerning documents representation & clustering

The first assumption we investigate is that Transformer-based language models are not adapted to the FSD algorithm and the task of dynamic clustering. Thus, we propose a new event detection system that treats the task of event detection on social media as a classical clustering task. To do so, we chose to discretize the stream using fixed-size windows. We experimented with both fixed-size windows (constant number of documents) and fixed time windows (constant duration of the window). The first advantage of this is that it ensures that documents clustered together have a similar publication date. Indeed, documents discussing the same event are likely to be posted in a similar period of time. Secondly, this allows using state-of-the-art clustering algorithms. In this context, we also fine-tuned language models to evaluate the interest of such an approach.

The second assumption we investigate is that representation models are complementary. They encode different parts of the documents and combining them can be profitable. We experimented with different ways of combining several representation models.

These problems are addressed in Chapter 4. Now, we move on to the next section in which we present the last phase of our event detection framework.

3.4 Phase 3 - Event identification

3.4.1 Description

The objective of this phase is to analyze the clusters of documents obtained during phase 2 to detect which clusters effectively deal with events. When those events are detected, they are compared to previously detected events, to determine whether they are new. Then, a summarization step is performed to present them in a friendly manner to a human.

To decide whether a cluster deals with an event, several aspects of the cluster can be considered, such as the diversity of the sources which posted the documents composing the cluster, or the diversity of the conversation in the group of documents.

As we stated before, we will discretize the stream by using disjoint windows. It allows us to use classical clustering algorithms but it adds the need of tracking the events across windows. Indeed, an event detected in a window does not necessarily stop right after the end of this window. Thus, once some clusters are labeled as events, if there are already some existing events that the system has detected, it is necessary to check whether a newly detected event refers to an already known event, as the continuation of it, or if it is truly a new event.

Finally, the summarization step is used to make the content of an event understandable for a human. This step is usually grouped with the tracking because the representation of a cluster is usually a representative document or representative named entities.

3.4.2 Problems raised concerning event identification

The first question is “how do you evaluate an event detection system”. Several annotated datasets exist, but it is difficult to evaluate correctly an event detection system, particularly on datasets extracted from social media such as Twitter. Indeed, the quantity of data is huge and it is not possible to annotate each document to know whether it is related to an event and if yes, which event. It raises some significant issues. First, to simulate a real-world setup using a dataset, we cannot only use documents annotated as event-related, because most of the documents posted only are not

event-related. Thus, using only annotated documents would create a bias. To avoid this bias, both event and non event-related documents must be considered. In this scenario, a new issue is that among the unannotated documents, some are related to annotated events, some are related to unannotated events, and some are not related to any events. Thus, evaluating systems based only on labeled documents is not sufficient when evaluating a full event detection pipeline.

The second question is "how to detect whether a cluster discusses an event". This is a critical point in an event detection framework as we can see from the literature and as we can infer from the name of this module.

The third question is "when an event is detected, how to determine whether it is linked to an event already known?".

Finally, how to give a meaningful summary of a cluster of documents to a human?

3.4.3 Solutions considered concerning event identification

First, we present different solutions for the event detection and the event tracking & summarization modules. we jointly study how to track and summarize events. We will develop an approach based on cluster chains such as (Fedoryszak et al., 2019). We will study different cluster representation, evaluate them and determine which representation is more suitable for the tracking of events.

Then, we present a way to compute classical evaluation metrics and new metrics to evaluate event detection systems on real-world setups. This problem was not clearly highlighted during the related work. We will devote a related work section to it in the chapter dealing with this problem. We consider that it is solely related to this part of the work and hence decide to separate it from the rest of the literature.

Finally, we compare our event detection system to other systems of the literature.

3.5 Conclusion

The rest of this thesis is organized as follows: In Chapters 4 and 5, we will present the Phase 2 and Phase 3 related problems. In Chapter 6, we will deal with the problems related to Phase 1. We chose this organization because Phase 2 and Phase 3 are more related to the core part of this thesis, the event detection method. Phase 1 is more closely related to the application of this work and is thus presented at the end of this document. Finally, in Chapter 7, we conclude our work.

Chapter 4

Data representation & clustering for event detection

In this chapter, we focus on the issues of the event detection framework related to Phase 2, namely related to the representation and the clustering of the documents. We introduce a part of our event detection system (EDS), carefully chosen to properly evaluate the modules of phase 2, and present the full system in the next chapter. We compare the performances of EDS performances with the First Story Detection (FSD) algorithm presented earlier in this document. Then, we compare, in the context of EDS, the performances of Transformer-based language models and TF-IDF. Finally, we study the combination of different combinations models in the context of this algorithm. Our main findings are that EDS performs better than FSD with every representation model, that language models are competitive with TF-IDF in our context, and that combining different representation models can be beneficial depending on the application.

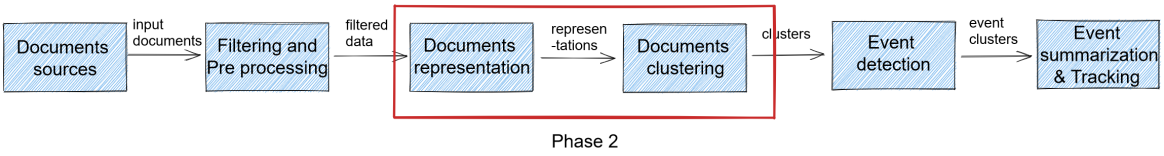

Figure 4.1: Phase 2 of the framework: Data representation and clustering

4.1 Introduction

This phase is composed of two main components: the Documents representation component and the Clustering component. Phase 2 can be considered as "the clustering phase" of the event detection framework, which is a crucial phase of such a framework. As we have seen in Section 2.4, nearly every event detection system uses a clustering approach, thus a good clustering is a condition for a good event detection system. The quality of the clustering depends on different factors: the content representation model, the similarity metric, and the clustering algorithm.

In this chapter, we introduce the first part of EDS, build on top of the event detection framework. EDS satisfies the constraints introduced earlier in the presentation. They are listed hereafter as a reminder:

- It must be an open-domain event detection system,
- It must be able to detect both large and small events,
- It must be able to use traditional clustering algorithms (in opposition to dynamic clustering).

The two first constraints come from the context we presented in the previous chapters. The last constraint is derived from the results obtained by (Mazoyer et al., 2020b): Transformer-based language models perform poorly in the context of FSD, which is a dynamic clustering algorithm. We want to experiment with whether these representation models perform better in the context of traditional clustering. Indeed, we believe that finding a configuration where Transformer-based language models perform the best is a good way to improve the overall performance of event detection, due to the current path that NLP research is following. Transformers are achieving the best performances across all the areas of NLP and the overall tendency of the domain is to consider as many features as possible when calculating content representation. This is one of the reasons why we chose to create a document-pivot model.

The rest of this chapter is organized as follows. First, we propose the first part of EDS and present it from a more practical point of view in Section 4.2. We compared the performances of EDS and FSD in several experiments and present these results in Section 4.3. We conducted this comparison for different text representation models, namely TF-IDF and Transformer-based language models.

Then, in the context of EDS, we compared the performances of these text representation models. We also experimented with whether it is interesting to fine-tune a language model for event detection, a task in which a lot of concept drift happens. This part is developed in Section 4.3.6.

Finally, we propose new document representation models, based on the combination of existing representation models. Indeed, we noted in the literature that text

representation models are usually individually used and are never used as complementary representation models that encode different aspects of the text content. This part is developed in Section 4.3.7.

In the next section, we introduce EDS and its different components. The work presented in this section was published in (Maître et al., 2021) and in a short paper to appear in RCIS 2022.

4.2 EDS: document representation & clustering

4.2.1 Description of the approach

We propose to treat the problem of event detection in textual data streams as a clustering task similar to that proposed by (Allan, 2012). This allows us to get out of the constraint imposed by dynamic clustering, i.e. we can thus consider all the documents published at the time of partitioning, and not have to work with fragmentary information over the flow of documents. We designed the method to be flexible, so any vectorial text representation model and any classical clustering algorithm can be used. This flexibility is interesting because it is important to be able to adapt the representation model/clustering algorithm pair, to adapt to the quickly evolving state-of-the-art of these domains. To be in a classical clustering context, we split the data stream using windows, i.e. fixed-size windows (fixed number of documents) or fixed time windows (documents published during a fixed period of time, i.e. 1 hour). This approach ensures that the documents clustered together have a close publication date, which improves the chances that the documents actually discuss the same event.

In this chapter, we are interested in evaluating the performances of different representation model/clustering algorithm pairs. To properly do that, we focus on the beginning of the framework presented in Figure 4.1, namely we stop after the "Documents clustering" step. Thus, we make the following hypothesis : (1) all the documents are event-related, (2) each document is associated with exactly one event, and (3) there is an unknown number of documents. Under these assumptions, we can reduce the framework and limit the steps that can affect the performance, and evaluate properly performances of each representation model. This is commonly done in the literature (Becker et al., 2010, Boom et al., 2016, Mazoyer et al., 2020a). No filtering will be performed on the documents as they are all event-related. In a more real-world setup, filtering steps are applied to filter spam and uninteresting documents. After the "Documents clustering" step, clusters are usually evaluated to determine whether they discuss an event or just a mundane conversation and then are summarized to be presented to humans. These steps are independent of the clustering phase in such a framework and thus are out of the scope of this Chapter. Considering these modifications, we present the adapted framework in Figure 4.2 and we will detail in a more

formal way each step of the process in the next section.

Figure 4.2: The framework considered in this chapter.

4.2.2 Formal description of the clustering process

First, we receive a stream of event-related input documents annotated as $D = \{d_1, ..., d_N\}$. We define a document as a $\forall i \in [1..N], d_i = (txt_i, dte_i, tag_i, url_i, src_i)$ where txt_i refers to the text content, dte_i to the publication date, tag_i refers to the tags and url_i refers to the urls shared and src_i refers to the source which posted the i^{th} document. We perform different cleaning steps described in Section 5.1 to obtain a set of cleaned documents. Then, we discretize the stream using fixed time windows (e.g. 1 hour) which is classical (Guille and Favre, 2014, McMinn and Jose, 2015, Naaman et al., 2011) because it is important to ensure that documents clustered together have a similar publication date, since documents dealing with the same events are usually posted during a similar period of time. They are annotated as $W = \{W^1, ..., W^m\}$ where $\forall k \in [1..m], W^k = \{d_1^k, ..., d_\tau^k\}$, where k refers to the k^{th} window and τ to the number of documents in each window. τ may be variable since each window is divided according to the time of publication and not the number of documents. Indeed, the number of documents posted varies over the course of the day. The windows are considered as independent from each others; i.e., $\forall k \in [1..m], \forall l \in [1..m], l \neq k, W^k \cap W^l = \emptyset$. Each window is partitioned in groups of similar documents known as clusters. The documents in W^k are then clustered according to similarity metrics (e.g. text similarity) to obtain a set of clusters such as $\forall i \in [1..n], \forall j \in [1..n], i \neq j, C_i^k \cap C_j^k = \emptyset$ and $\bigcup_{j=1}^{n} C_j^k = W^k$.

Thus, our event detection system is a succession of clustering processes as a result of the discretization of the stream using fixed and disjoint time windows. This differs from the FSD algorithm which treats the problem of event detection as a dynamic clustering problem. We will now present the different algorithms and models used for each step. A more visual description of the process for a window is proposed in Figure 4.3 and a pseudo-algorithm is provided in Algorithm 3.

4.2.3 Algorithms and models considered

We propose to compare different text representation models in two different contexts: FSD, presented in more detail in Section 2.4.1, and EDS. For both of these contexts, we will perform three majors steps, i.e. text representation, similarity calculation

Algorithm 2: EDS, Clustering Part

input : threshold t, window W
output: L, a list of clusters for window W

$Repres \leftarrow []$; $SimMatrix \leftarrow []$; $ListClusters \leftarrow []$;
foreach *document d in W* **do**
  $Repres(d) \leftarrow RepresentationModel(d)$;
end
for (d_1, d_2) *in W* **do**
  $SimMatrix(d_1, d_2) \leftarrow Cosine(Repres(d_1), Repres(d_2))$
end
$ListClusters \leftarrow ClusteringAlg(SimMatrix, t)$;

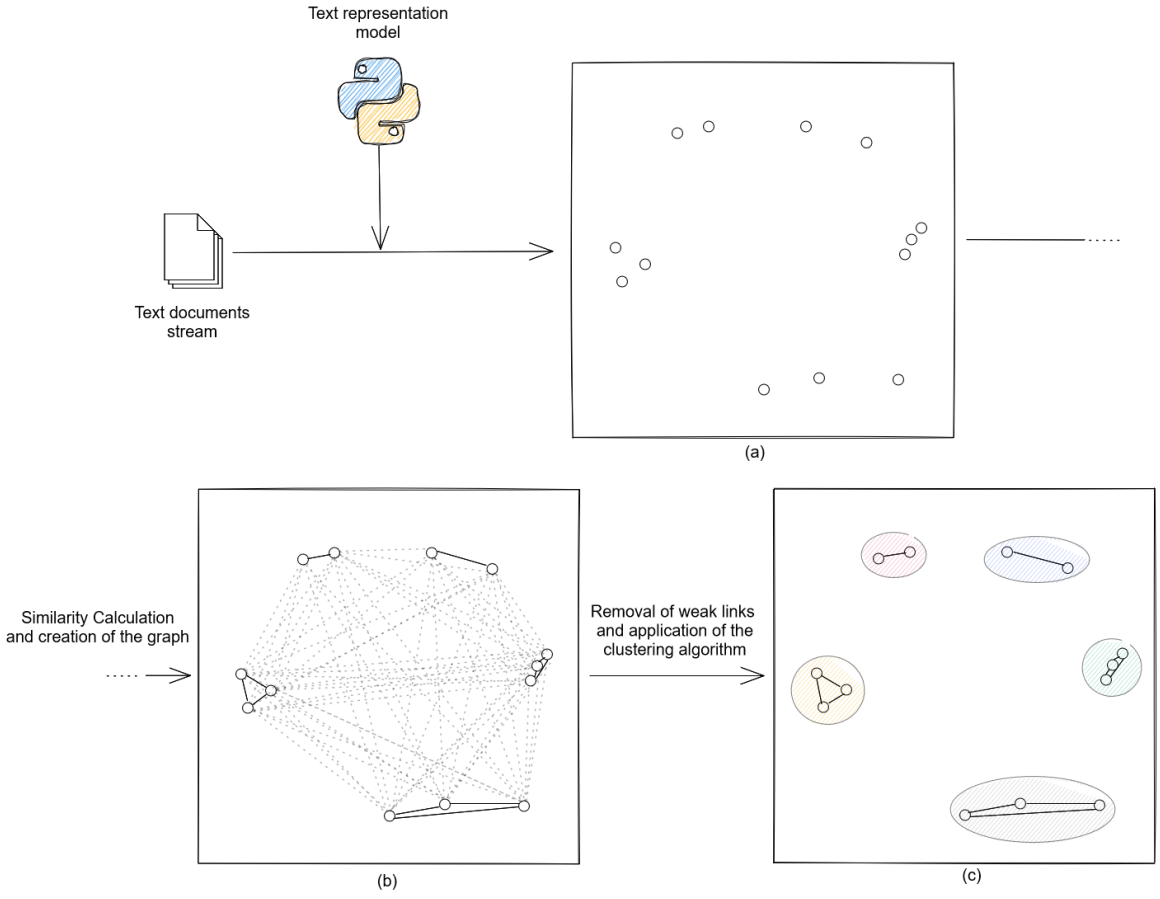

Figure 4.3: Data treatment process performed by EDS for each window. (a) Documents representations in vector space. Each document is represented by a point. (b) A graph is created using the similarity matrix. Each document is a vertex and each edge is weighted using the similarity between documents. (c) Creation of the clusters, by deleting edges with a low weight.

between documents, and clustering. We will first present the representation models and then the similarity calculation and clustering.

Representation models

We compare two types of text document representations: statistical approaches, also called lexical approaches and Transformer-based language models, also called semantic approaches.

Lexical approaches - We use TF-IDF, which is the most common text document representation model in information retrieval (Baeza-Yates et al., 1999). It represents the importance of a word in a document based on its frequency in it and its frequency in the whole corpus, under the assumption that a word that is frequent in a document but not in the corpus is representative of the document. We use an IDF calculated on the whole dataset Event2012 (McMinn et al., 2013) in section 5.1, provided by (Mazoyer et al., 2020a) and do not take into account term-frequency (TF) because most of the word appears only once in short documents.

Semantic approaches - Semantic representations of text documents are currently the state-of-the-art in NLP, particularly using Transformer-based language models (Vaswani et al., 2017). In particular, we will compare two languages models: S-BERT (Reimers and Gurevych, 2019) and Universal Sentence Encoder (USE) (Cer et al., 2018).

Clustering

For each pair of documents and each document representation model, we compute its similarity to constitute a similarity matrix S_{model,W_k} used to compute the clusters. We chose Cosine Similarity as it is the most common similarity measure in NLP (Aggarwal and Zhai, 2012) and is recommended with embedding vectors produced using Transformer models (Reimers and Gurevych, 2019). It is important to note that the performances of the clustering are directly affected by the similarity measures making it a critical step of the event detection process.

Using these similarities, clusters are computed using the Louvain algorithm (Blondel et al., 2008), a well-known community detection algorithm that automatically computes the optimal number of clusters. This aspect is especially important in our context of open-domain event detection, in which the number of events is not known beforehand. The only parameter that this algorithm need is a similarity threshold, determined by optimization, which will be different for each representation model.

Now that we have presented the different algorithms we use, we present the dataset on which we conducted most of the experiments of this thesis.

4.3 EDS and FSD : experiments and results

In this section, we present different experiments. First, in a preliminary study, we examine the impact of the type and size of the window on the performances. Then, we compare EDS and the FSD algorithm. In the next two experiments, we compare the performances of different text representation models. The goal of the second experiment is to evaluate the performances of Transformer-based language models compared to TF-IDF in the context of EDS. Then, in the third experiment, we evaluate the utility of the fine-tuning of the Transformer-based language models.

For each of these experiments, we will first present the experimental protocol and then the results. We will include significances tests, using $\alpha = 0,05$. We evaluate the significance of the test using the "Wilcoxon signed-rank test", which is the method that fits the best in our context (Yeh, 2000). Indeed, we use non-parametric test methods due to the characteristics of our data.

4.3.1 Dataset

We use Event2012 (McMinn et al., 2013), a corpus of 120 million tweets, collected from the 10th of October to the 7th of November 2012 from the Twitter streaming API. 159,952 tweets are labeled as event-related, distributed into 506 events, which are distributed into 8 categories. Some details about the dataset are illustrated in figure 4.4, 4.5, 4.6 We only work on the annotated part of the dataset in order to be able to evaluate properly our results. Due to the policy of Twitter, only tweet ids can be shared and the actual content of the tweets has to be retrieved using the Twitter API. Some tweets are not available anymore, due to deletion of the tweet, of the account which posted the tweet, or because the account is not public anymore. Thus, we collected 69,875 labeled tweets, which are distributed into 504 events. To simulate a stream of data as it would be in a real-world context, we sorted the dataset according to the date of publication of each tweet. We divide the dataset into 3 equal sets: the training set, the validation set, and the testing set, each of them composed of 9 days of data.

4.3.2 Experimental configuration

Evaluation measures

B-cubed is a generalization of Precision, Recall, and F1-score for clustering and is the most complete cluster evaluation measure (Amigó et al., 2009). Precision P is defined as the proportion of documents in the document's cluster that correspond to the same event. The corresponding equation is equation 4.2. Recall R is defined as the proportion of documents that correspond to the same event, which are also in the

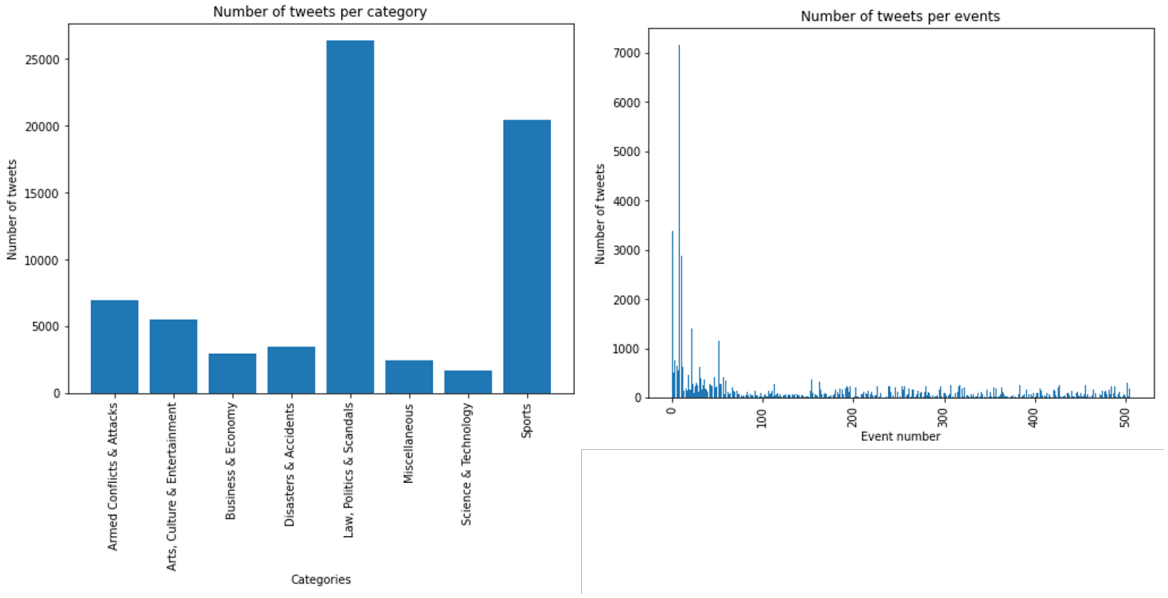

Figure 4.4: The number of tweets per category and events. In both cases, the repartition of tweets is not homogeneous.

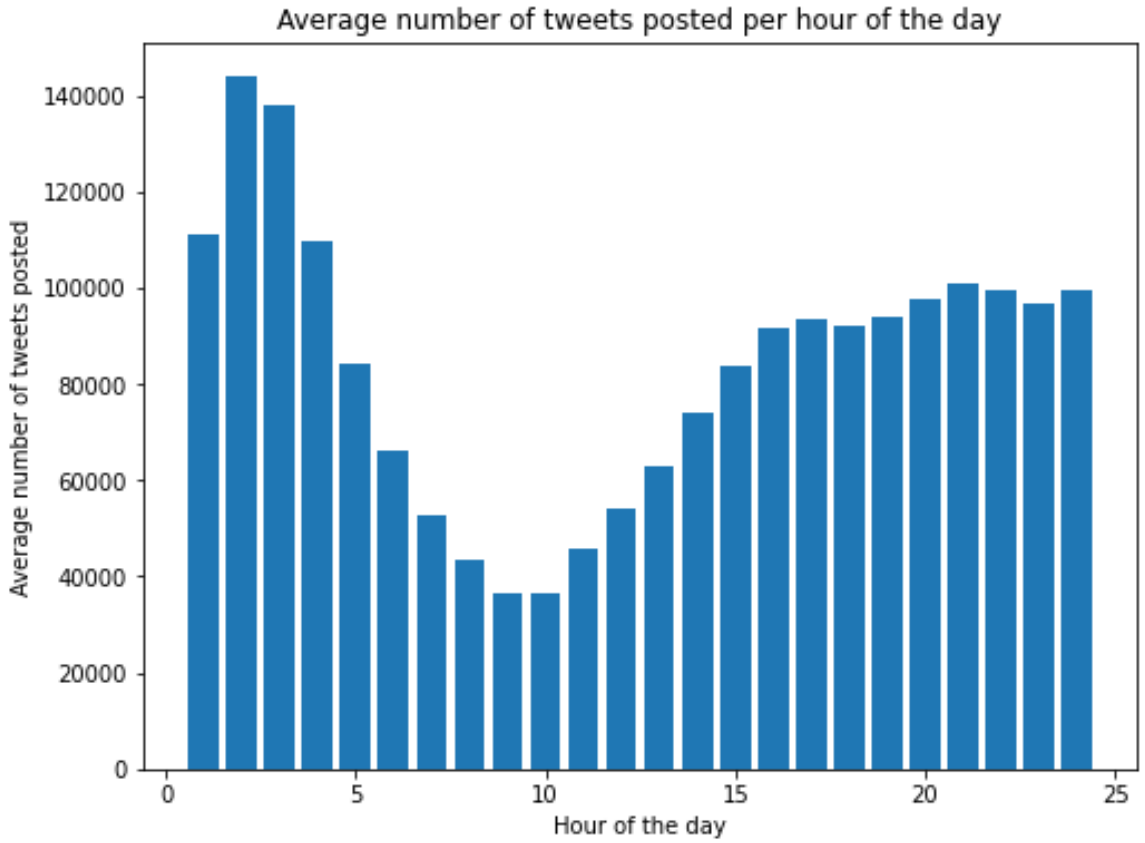

Figure 4.5: The average number of tweets posted per hour of the day (UTC).

Event Number	Event Description	Number of tweets
8	During US presidential debate, President Barack Obama tells candidate Mitt Romney he is "the last person to get tough on China.	7154
1	12 Oct 2012 " Paul Ryan spoke for 40 of the 90 minutes during Thursday night's vice presidential debate and managed to tell at least 24 myths during that time	3380
11	Barack Obama And Mitt Romney Went Head-To-Head In The Final Presidential Debate romney said not government that makes businesses successful!	2871
157	They were discussing about Rondo	1551

Figure 4.6: Some examples of events. We chose the events with the highest number of labeled tweets.

document's cluster. The corresponding equation is equation 4.3. B-cubed is illustrated in Figure 4.7 and precision and recall are computed using the following way. Let $L(e)$ and $C(e)$ denote the category and the cluster of an item e. The correctness of the relation between e and e' is defined as:

$$Correctness(e, e') = \begin{cases} 1, & \text{iff } L(e) \leftrightarrow C(e) = C(e') \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

$$Precision = Avg_e(Avg_{e'.C(e)=C(e')}(Correctness(e, e')) \tag{4.2}$$

$$Recall = Avg_e(Avg_{e'.L(e)=L(e')}(Correctness(e, e')) \tag{4.3}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}. \tag{4.4}$$

Split of the dataset

To conduct our experiments on a dataset as close to reality as possible, we order the documents in chronological order and split them into windows. It is a particularly important parameter for the training phase of the S-BERT model, which is detailed next. Indeed, the vast majority of the event labels that are present in the training set are not in the test set. The training set is constituted of 225 events, while the test set is constituted of 303 events. There are 24 common events in these sets. This is illustrated in Figure 4.8

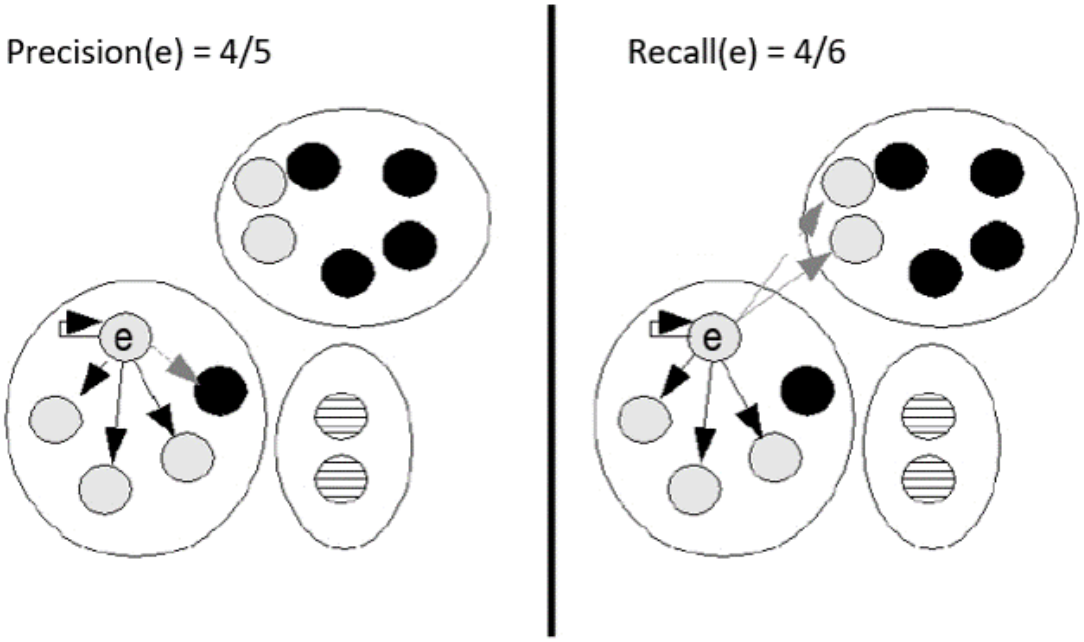

Figure 4.7: Example of computing the BCubed precision and recall for one item. Figure extracted from (Amigó et al., 2009). In our case, a circle corresponds to a document, a color to a ground truth event, and a bubble to a cluster.

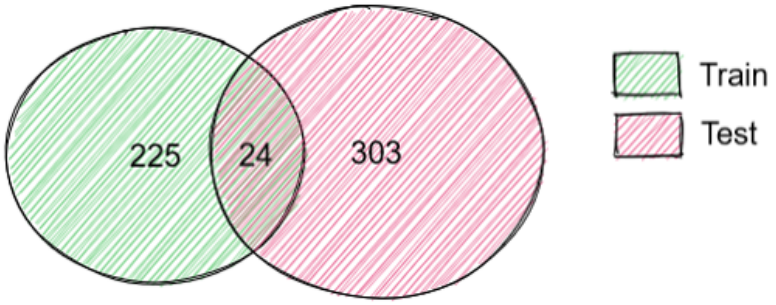

Figure 4.8: Repartition of events between the training set and the test set. Only a few events are in common, due to the drift happening in the conversations.

Representation models

In this experiment, we propose two variations of TF-IDF and S-BERT, and we use the model USE-LARGE[1], which will be called USE in the rest of this experiment. Concerning TF-IDF, we use the implementation proposed by (Mazoyer et al., 2020b). The first one, named **TF-IDF dataset**, calculated IDF on the labeled tweets of the dataset. The second, **TF-IDF all tweets**, calculated IDF on the whole dataset. Concerning S-BERT, the first version, named **S-BERT nli** est the pre-trained version on the NLI dataset, and is available using the implementations proposed by the authors of (Reimers and Gurevych, 2019)[2]. This model is based on a siamese network, composed of two equal BERT models. We chose this BERT model because the SNLI dataset is known to improve the performances of the models for clustering tasks (Bowman et al., 2015). The second version of S-BERT is **S-BERT fine-tuned**, is a fine-tuned version of S-BERT on the training set, which is the first half of the labeled dataset. The events are used as the target labels. The particularity of this training set is it is ordered according to the publication date of the documents, thus, the major part of the event in the training set is not in the test set, as we said earlier. We assigned to each tweet a pair of tweets, a tweet from the same label, and a tweet from a different label, as it is usually done to train siamese neural networks. Each of these two tweets is randomly chosen in the training set.

4.3.3 Preliminary experiment: impact of the window

Experimental Protocol

The objective of this experiment is to compare different ways to discretize the stream. It is an important parameter of our model because it directly impacts the clustering results. Our intuition is that documents talking about the same events are posted in a short period of time. Thus, we compare the performances of different text representation models with different values of windows in the context of EDS. We experiment with two types of windows: temporal windows, and fixed number of documents windows. Each type of window has its advantages: the fixed time window ensures that documents are posted in a short and related period of time. However, some windows can be nearly empty due to the variation in the number of tweets in the day, making the events detected potentially irrelevant. Concerning fixed-size windows, they allow for anticipation of the memory usage of the algorithm. It can be important in some cases when running on a machine with limited memory resources.

In this experiment, we use 6 different windows and compare the results. We use time windows of 1 hour, 2 hours, and 4 hours. We use fixed-size windows of 1000

[1]https://tfhub.dev/google/universal-sentence-encoder-large/5
[2]https://github.com/UKPLab/sentence-transformers

tweets, 2000 tweets, and 4000 tweets.

Results

The results are displayed in Table 4.1. We begin with the analysis of the fixed-size windows. As we can see, the F1-score is relatively stable when the number of tweets per window varies. However, the precision and recall values are not stable: when the number of tweets per window increases, the precision decreases while the recall increases. When working with hours windows, the results are stable. This is probably because there are fewer differences induced by the variation of the duration of each window than by the variation of the number of tweets per window. The labeled tweets are not equally distributed in time and a variation of a thousand tweets is proportionally big compared to the size of the dataset.

Overall, the performances are better when the stream is discretized using time windows. Even if it implies that each window will have a different number of tweets and thus hold a different space in memory, we think it is a better discretization method, which is more coherent with what is usually done when reporting information. Indeed, it is usual to hear that there is an hourly update of the news when something is happening. Keeping in mind the idea of the downstream task which is to present important events to buyers, an hourly report is more natural than a report based on the number of tweets posted. Since the differences between time values are not significant, we will perform our next experiments using 1 hour time windows, because we believe it is a better granularity to have the unfolding of events.

4.3.4 Comparison of EDS and FSD

This first experiment is the comparison of the four text representation models, **TF-IDF dataset**, **TF-IDF all tweets**, **S-BERT nli** and **USE**, in two different contexts, i.e. in the context of FSD or in the context of EDS.

Experimental protocol

For the FSD implementation, we use the one proposed by (Mazoyer et al., 2020a)[3]. Thus, we formulate the following H0 hypothesis: “There is no statistically significant difference between the performance of the algorithms in the FSD and EDF”. To validate this hypothesis, we use the “Wilcoxon signed-rank test”.

Experimental configuration

Concerning the threshold values used for the FSD algorithm, we used the same as the one presented in (Mazoyer et al., 2020b), i.e. t=0.65 for TF-IDF dataset, t=0.75 for

[3]https://github.com/ina-foss/twembeddings

Table 4.1: Clustering quality according to the metric B-Cubed for each textual representation, depending on the size of the window. Time windows seem to be more adapted.

Window	Model	Precision	Recall	F1 Score
1000 tweets	TF-IDF dataset	0.81 ± 0.10	0.74 ± 0.30	0.71 ± 0.20
	TF-IDF all tweets	0.81 ± 0.10	0.74 ± 0.30	0.72 ± 0.20
	USE	0.82 ± 0.12	0.76 ± 0.27	0.74 ± 0.17
	SBERT	0.95 ± 0.04	0.35 ± 0.20	0.48 ± 0.22
2000 tweets	TF-IDF dataset	0.78 ± 0.11	0.76 ± 0.27	0.72 ± 0.19
	TF-IDF all tweets	0.78 ± 0.12	0.76 ± 0.27	0.71 ± 0.19
	USE	0.76 ± 0.13	0.80 ± 0.25	0.73 ± 0.15
	SBERT	0.92 ± 0.05	0.38 ± 0.19	0.51 ± 0.20
4000 tweets	TF-IDF dataset	0.72 ± 0.11	0.79 ± 0.26	0.70 ± 0.14
	TF-IDF all tweets	0.72 ± 0.12	0.78 ± 0.26	0.70 ± 0.14
	USE	0.69 ± 0.15	0.81 ± 0.24	0.70 ± 0.12
	SBERT	0.89 ± 0.06	0.41 ± 0.16	0.53 ± 0.16
1 hour	TF-IDF dataset	$\mathbf{0.84} \pm 0.09$	0.80 ± 0.21	$\mathbf{0.80} \pm 0.13$
	TF-IDF all tweets	$\mathbf{0.84} \pm 0.09$	0.80 ± 0.21	$\mathbf{0.80} \pm 0.13$
	USE	$\mathbf{0.91} \pm 0.08$	0.86 ± 0.19	$\mathbf{0.87} \pm 0.12$
	SBERT	$\mathbf{0.97} \pm 0.05$	$\mathbf{0.56} \pm 0.22$	$\mathbf{0.68} \pm 0.19$
2 hours	TF-IDF dataset	0.82 ± 0.08	0.82 ± 0.19	0.80 ± 0.12
	TF-IDF all tweets	0.81 ± 0.08	0.81 ± 0.19	0.80 ± 0.12
	USE	0.88 ± 0.08	$\mathbf{0.87} \pm 0.17$	0.86 ± 0.11
	SBERT	0.96 ± 00.5	0.51 ± 0.19	0.64 ± 0.18
4 hours	TF-IDF dataset	0.80 ± 0.08	$\mathbf{0.84} \pm 0.19$	0.80 ± 0.12
	TF-IDF all tweets	0.80 ± 0.08	$\mathbf{0.84} \pm 0.19$	0.80 ± 0.12
	USE	0.84 ± 0.08	0.87 ± 0.16	0.84 ± 0.10
	SBERT	0.95 ± 0.04	0.47 ± 0.17	0.61 ± 0.17

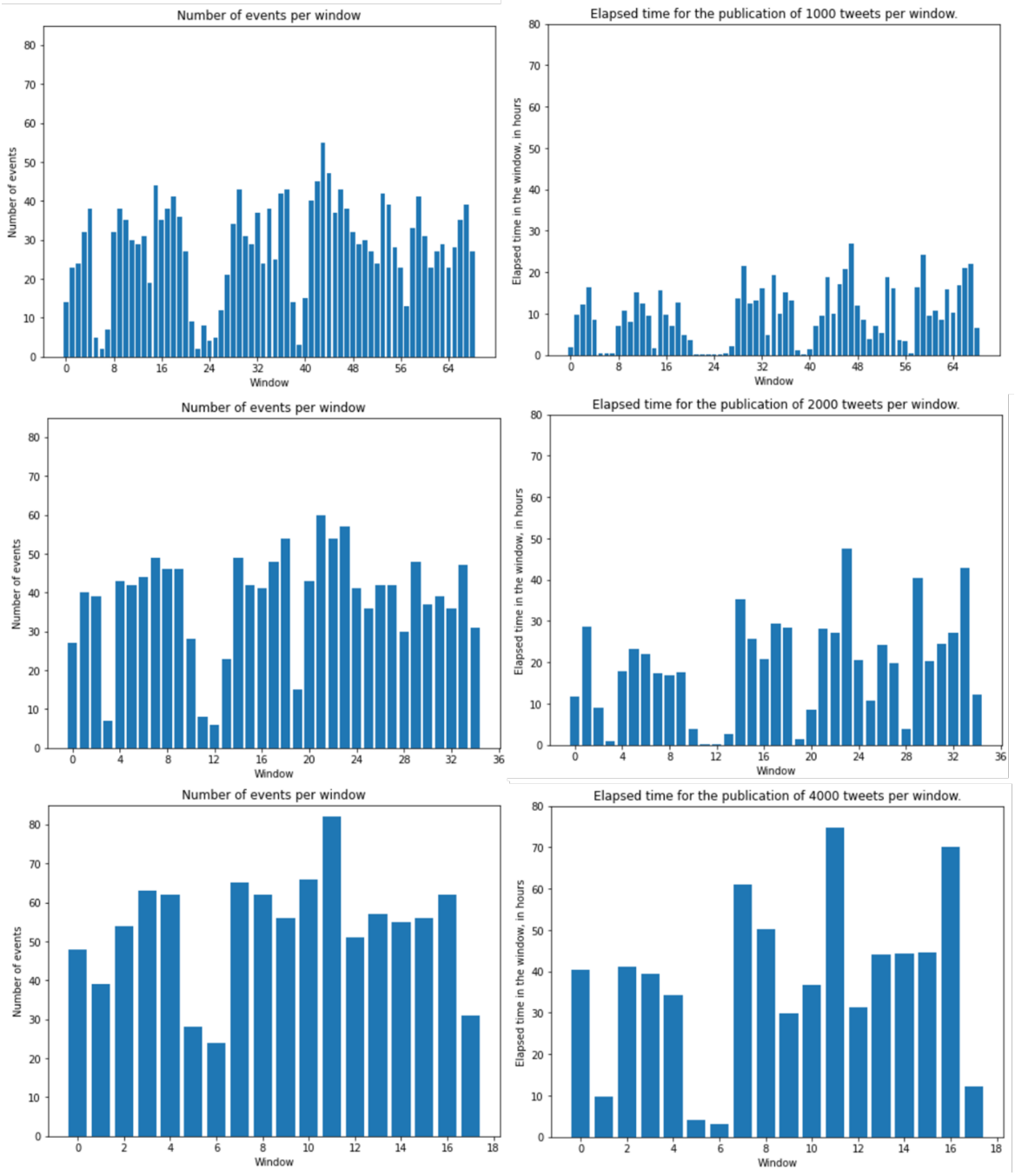

Figure 4.9: The number of events per fixed-size windows and elapsed time for the publication of the defined number of tweets. As we can see, for each fixed number of tweets, there is a lot of disparity between the windows. However, we can see similar characteristics when the number of tweets varies.

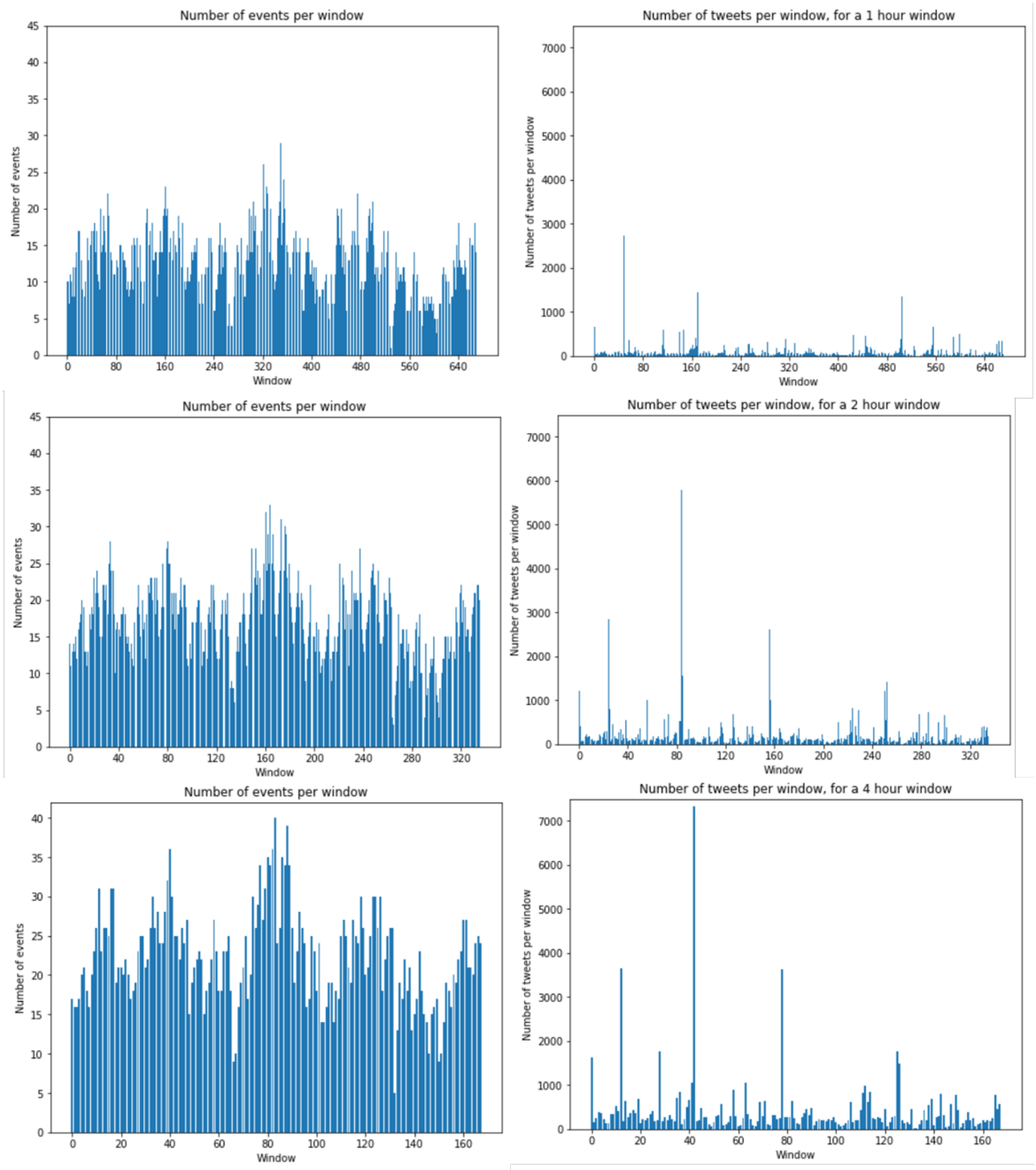

Figure 4.10: The number of events per time window and the number of tweets per time window. As we can see, there is a lot of disparity between the windows. However, we can see similar characteristics when the elapsed time varies.

Table 4.2: Clustering quality according to the metric B-Cubed for each textual representation, according to the clustering algorithm. In every case, EDS performs better than FSD. We display the EDS results with a ± value because it is the mean of the value across all the windows. FSD is a single evaluation.

Model	Approach	Precision	Recall	F1 Score
TF-IDF dataset	FSD	0.70	0.59	0.64
	EDS	**0.84** ± 0.09	**0.80** ± 0.21	**0.80** ± 0.13
TF-IDF all tweets	FSD	0.82	0.50	0.62
	EDS	**0.84** ± 0.09	**0.80** ± 0.21	**0.80** ± 0.13
USE	FSD	0.85	0.38	0.52
	EDS	**0.91** ± 0.08	**0.86** ± 0.19	**0.87** ± 0.12
S-BERT-nli	FSD	0.95	0.31	0.46
	EDS	**0.97** ± 0.05	**0.56** ± 0.22	**0.68** ± 0.19

TF-IDF all tweets, t=0.39 for S-BERT and t=022 for USE. The threshold values used for EDS are the following: t=0.10 for models based on TF-IDF, t=0.80 for S-BERT, and t=0.60 for USE. As a reminder, these similarity values are computed using Cosine Similarity. These threshold values were determined empirically.

Results

Table 4.2 shows the results of this experiment. The numbers presented are the mean of each metric for each window and the standard deviation. Because of the nature of FSD, presented in section 2.4, we did not use time windows and obtain a single measure for each metric. In every case, EDS performs better than FSD, particularly in terms of recall.

4.3.5 Comparison of text representation models

The second experiment consists in the comparison of **TF-IDF dataset**, **TF-IDF all tweets**, **S-BERT nli** and **USE** in the context of EDS. This experiment is useful to compare these representation methods to each other, to determine which is the most efficient method. In particular, we want to investigate the relative performances of the Transformer-based language models compared to the models based on TF-IDF. As a reminder, in Mazoyer et al. (2020a), they showed that the Transformer-based language models were poorly performing on this dataset in the context of the FSD algorithm and that the models based on TF-IDF performed the best

Experimental protocol

We evaluate the performances using the B-cubed metric and formulate the following H0 hypothesis: “There is no statistically significant difference between TF-IDF based

Table 4.3: P-value for the Wilcoxon signed-rank test, to compare each text representation model. In every case, P-value$<\alpha$.

	Precision	Recall	F1 Score
S-BERT nli / TF-IDF dataset	1.25e-100	5.54e-79	2.78e-49
S-BERT nli / TF-IDF all tweets	7.08e-101	1..47e-79	8.59e-50
USE / TF-IDF dataset	1.24e-70	5.70e-34	3.39e-77
USE / TF-IDF all tweets	6.24e-72	1.15e-33	3.96e-77

models and Transformers-based models".

Experimental configuration

The threshold values are the same as the previous experiment, i.e. t=0.10 for models based on TF-IDF, t=0.80 for S-BERT, and t=0.60 for USE.

Results

We compare each method by applying them to the previously defined windows, in an unsupervised context because none of the models used the event labels during a training phase. The results are shown in Table 4.2, for the lines corresponding to EDS. The results of the significance tests are presented in Table 4.3.

Thus, the performance is on average better for TF-IDF based approaches compared to S-BERT in terms of recall, while S-BERT performs better in terms of Accuracy. USE performs better on all metrics. The significance tests have a p-value < 0.05. We can therefore reject the H0 hypothesis that was formulated, and conclude that the differences are significant.

4.3.6 Fine-tuning in the context of event detection on social media

For this experiment, we complement the structure with a few additional toy experiments to ease the understanding of the intuition behind the goal of the experiment. While it might be obvious to think that fine-tuning a Transformer-based language model for a task will improve the performance of the model, it is not so straightforward in a context of concept drift. As we saw earlier, most of the target events of the test sets do not exist in the training set. Thus, if the model actually learns from the training phase, it could be understood that the model can generalize some of the information it learned during this training phase. Before conducting the actual experiment, we conducted a first toy experiment where we split the dataset in two, without taking into account the order of publication of the documents. Then, we visualized the representation obtained for the test set, using the t-SNE method. We show the results of this experiment in figure 4.11. Then, we conducted the same experiment

using this time the time-ordered dataset. Even if the performances are worse, there is still a notable improvement compared to the configuration without fine-tuning. Thus, we decided to conduct the following experiment.

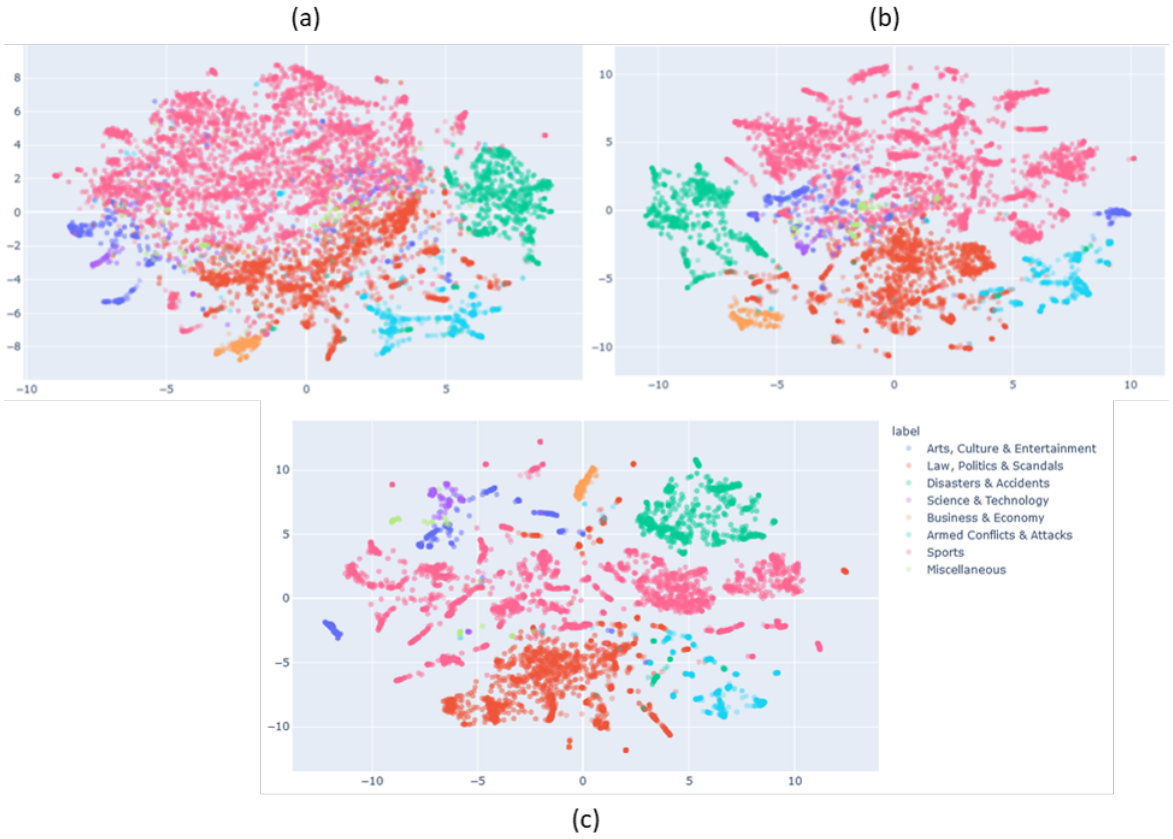

Figure 4.11: t-SNE representation of the S-BERT embeddings of the documents from the test set, in three configurations: (a) without fine-tuning, (b) fine-tuning on time-ordered set, (c) fine-tuning on half of the dataset, chosen randomly. As we can see, even if the groups of documents seem to approximately be regrouped by category in (a), it does not seem they are creating different clusters for each event. In the two other images, clusters seem more obvious can could correspond to events. As can be expected, training on random data is more efficient than training on time-ordered data, as the training set is more representative of what will be encountered on the test set. However, it is not a realistic scenario. Training on the training set in a time-ordered manner still seems to be beneficial, explaining why we decided to conduct this experiment.

Experimental protocol

We compare **TF-IDF dataset**, **TF-IDF all tweets**, **S-BERT fine-tuned** and **USE** in the context of EDS, on the test dataset. This experiment is similar to the previous one but the goal is different. Here, we want to validate whether fine-tuning the S-BERT model is interesting in a context of a data stream. To be coherent with this context, we trained the S-BERT model on the training set, which has been presented earlier, in section 4.3.2. We did not fine-tune USE because it cannot be easily done in the

Table 4.4: Clustering quality according to the metric B-Cubed for each textual representation, in a supervised context, on the test dataset.

	Précision	Rappel	F1 Score
TF-IDF dataset	0.83 ± 0.10	0.79 ± 0.20	0.79 ± 0.13
TF-IDF all tweets	0.83 ± 0.10	0.79 ± 0.20	0.79 ± 0.13
USE	0.90 ± 0.08	$\mathbf{0.86} \pm 0.18$	$\mathbf{0.86} \pm 0.12$
S-BERT fine tuned	$\mathbf{0.95} \pm 0.06$	0.77 ± 0.17	0.83 ± 0.12

Table 4.5: P-value for the Wilcoxon signed-rank test. Not all the results are significant, notably for F1 Score of S-BERT and TF-IDF.

	Précision	Rappel	F1 Score
USE / TF-IDF	8.77e-37	3.49e-22	3.80e-43
S-BERT nli fine-tuned / TF-IDF	1.99e-54	5.74e-06	2.35e-19

current version of the model. Anyway, BERT is currently the most standard language model, so it is logical to focus on this particular language model. We still apply USE to the training set to compare the results.

Experimental configuration

The performances are evaluated using B-cubed. We formulate the following H0 hypothesis: "None of the approaches is significantly better than the others". The threshold values are the same as the previous experiments, i.e. t=0.10 for models based on TF-IDF, t=0.80 for S-BERT, and t=0.60 for USE.

Results

Results are presented in Table 4.4 and the results of the significance tests in Table 4.5.

We can see that the results are significantly better for the Transformers architectures compared to the TF-IDF approaches.

4.3.7 General discussion of the results

The first experiment showed that EDS performs better than the FSD algorithm in most of the presented cases. This finding is especially true for the recall measure. Concerning precision, and particularly for Transformer-based language models, the values of FSD and EDS are close. We believe that the FSD algorithm allows in these cases to obtain coherent clusters (high precision). However, it seems that the FSD tends to segment documents of the same label in different clusters, resulting in a drop in recall. This is probably because the FSD algorithm can create a new cluster when a

new document arrives, without taking into account all of the documents in the window. This segmentation is less frequent with EDS, explaining the better recall values.

We also showed that the Transformer-based language models, especially USE and S-BERT fine-tuned, can be competitive with classical methods (TF-IDF). We can note that in an unsupervised context, S-BERT performs worse than USE. We believe this is due to the dataset used for the pre-training of the different language models. Indeed, the S-BERT model that we used is based on BERT NLI, which is trained on the English Wikipedia Corpus, on BookCorpus, and fine-tuned on SNLI. USE is, for its part, trained on a more diverse dataset, including data from discussion forums, and question-answer websites. These data are closer to the one we encounter in the dataset Events2012, which is extracted from Twitter. Thus, data extracted from social networks, for which the syntax is very specific because of the destructuration of the language, are a problem for the vanilla S-BERT because it is trained on data written in more conventional English. Once S-BERT is fine-tuned on social network data, the performances rise and they become similar to the performances of other models. Thus, the fine-tuning phase is particularly important and it shows that fine-tuning S-BERT on data extracted from social networks allows us to obtain better results in our context.

4.3.8 Partial conclusion

In this section, we showed that considering the problem of event detection as a clustering problem (EDS) rather than a dynamic clustering problem (FSD) allows us to achieve better performances. We also showed that in a certain context, Transformer-based language models can have performances similar to classical models (TF-IDF). Finally, we showed that the fine-tuning of these language models are particularly interesting to adapt to the specific data extracted from the social networks. Now that we showed that different text representation models can be interesting to represent the documents, we would like to explore whether combining them can be interesting for document representation. More specifically, we want to combine lexical representation models (TF-IDF) and semantic representation models. Moreover, there are several other aspects of the documents that can be interesting to represent and compute their similarity. We will explore this in the next section.

4.4 Combination of models

As we have seen in the previous section, grouping documents dealing with the same event is a challenging task. As highlighted in Section 2.4, most of the event detection approaches are based on text representation models to represent the content of the documents. However, data stream documents are composed of several features such

as text, indexes (e.g. hashtags), metadata, images, links, and social media-specific features (repost, user mentions...). To fully comprehend the content of the documents, all these features can be interesting (Spina et al., 2014).

Thus, in this section, we investigate the combination of different document content representation models in the context of EDS. These models exploit either lexical, semantic, or social network-specific features to represent the documents. To the best of our knowledge, these models are usually individually exploited in event detection methods and not combined as models encoding different aspects of the documents. We base this assumption on the analysis of the literature, which is summarized in Table 4.6. To encode lexical features, we use TF-IDF, and to encode semantic features, we use Universal Sentence Encoder, as we have seen in the previous section that it performs the best in our context. In terms of specific features, we consider tags and URLs shared by users, features classically used for event detection (Hasan et al., 2018). Our goal in this section is to investigate whether the information encoded by each of these models is complementary for the task of clustering event-related social network documents. To do so, we first evaluate the performances of each of these individual models to perform this task on a training set, in a similar way to the previous section, to weight each of the models in the combination. Then, we explore combinations of these models to obtain new similarity measures between documents used in the clustering task. To evaluate our models and conduct our experiments, we focus on Twitter which is the most commonly used social network in research, and use Event2012 (McMinn et al., 2013) as in the previous section.

We evaluate these models in the context of EDS, as in the previous section. The only difference is that we introduce a new type of feature, social network-specific features, described as follows :

Social network-specific features - Hashtags are inherent to the Twitter ecosystem and this mechanism has now spread to other social networks and even newsfeeds and is now used on most of them. They help classify documents and assign them to the right feed. Thus, it is clear that including them to determine social media document similarity is important (Morabia et al., 2019). Another interesting feature is links shared by users. Most of the time, social media documents such as tweets are used to react to some news which are also discussed on other websites such as press websites. Documents containing the same link could discuss a similar subject. Links have proven to be important in some recent work on event detection (Hasan et al., 2019), (Quezada and Poblete, 2019)./-*

To encode the similarity for these new features, we use Jaccard-Dice similarity to determine which documents have common tags and we determine if documents are sharing the same URL by simple string comparison. The same clustering algorithm is used, namely the Louvain algorithm (Blondel et al., 2008).

Table 4.6: Comparison of different clustering-based event detection approaches from the literature

Article	Mostly Based on text	Combination of Features	Type of features considered	Text Representation method	Aspect of the text considered
(Petrović et al., 2010)	Yes	No	Text	TF-IDF	Lexical
(Hasan et al., 2019)	Yes	No	Text	TF-IDF	Lexical
(Mazoyer et al., 2020a)	Yes	Yes	Text	IDF, USE, SBERT, BERT	Lexical or Semantic
(McMinn and Jose, 2015)	Yes	No	Text	TF-IDF & Named Entities	Lexical
(Naaman et al., 2011)	Yes	Yes	Text, Twitter Specific	TF-IDF, boosted weight on hashtags	Lexical
(Boom et al., 2016)	Yes	Yes	Text	Tf-IDF weighted Word2vec	Semantic
(Zhou et al., 2017)	Yes	No	Text	Word Embeddings	Semantic
(Li et al., 2017)	Yes	Yes	Text	Word Embeddings	Semantic
(Becker et al., 2010)	No	Yes	Text, Time, Location	TF-IDF	Lexical
(Cai et al., 2015)	No	Yes	Text, Image, Time, Location, Hashtags	Topic Modelling	Diverse
(Guille and Favre, 2014)	No	Yes	Text, Twitter specific	Word cooccurences	Lexical
(Han et al., 2019)	No	No	Location	/	/

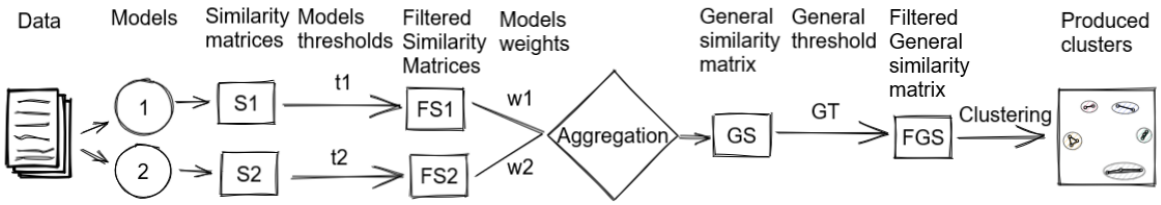

Figure 4.12: Illustration of the method for the combination of two representation models.

4.4.1 General description of the combination method

We combine different representation models using an ensemble-based similarity (Domeniconi and Al-Razgan, 2009, Gionis et al., 2007, Strehl and Ghosh, 2002), a classical approach to combine different clustering methods that have also been used in related work (Becker et al., 2010). The principle of the technique is to jointly exploit the results of different clustering processes to take advantage of their combined strengths. In our case, we have two main propositions. First, we propose to combine text representation models that encode different aspects of the text, which is not done in the literature. We also complement this representation with social-media-specific features that are classically used in the literature. Second, we propose to combine the results at the similarity level, to have more flexibility for the aggregation of the results (presented in Section 4.4.2). The process is illustrated in Figure 4.12. For each representation model, a similarity matrix is computed. A model threshold is applied to this matrix to filter low similarities. Then, each similarity matrix is weighted according to the performance of its respective model for the clustering task. The matrices are aggregated to obtain a general similarity matrix, which is filtered using a general threshold. This filtered matrix is then used for the clustering.

The different aggregation methods, configurations, and how to obtain the thresholds and weights are described in the next sections.

4.4.2 Aggregation methods

One of the main steps of our method is to aggregate similarity matrices together. We propose different aggregations and compare their performances in the result section.

Similarity aggregation (SA): For each representation model, if the pairwise similarity is superior to its optimal threshold, then this similarity value is used in the weighted sum. Here is an example with three representation models with respective weights of 0.50, 0.35, and 0.15. The two first representation models find a pairwise similarity of respectively 0.74 and 0.86 which are above their optimal threshold. The last representation model finds a value under its optimal threshold, so the value is set to 0. The overall pairwise similarity will be 0.50*0.74

+ 0.35*0.86 + 0.15*0 = 0.671. More formally, for each a_{ij} of S, the similarity matrix of a model:

$$fa_{ij} = \begin{cases} 0, & \text{if } a_{ij} < t \\ a_{ij}, & \text{otherwise} \end{cases} \tag{4.5}$$

Binary aggregation (BA): Instead of transmitting the similarity value, we compute whether the value is above the threshold. If yes, the value 1 is transmitted to the weighted sum. If no, the value 0 is transmitted. More formally:

$$fa_{ij} = \begin{cases} 0, & \text{if } a_{ij} < t \\ 1, & \text{otherwise} \end{cases} \tag{4.6}$$

General Aggregation (GA): We compute the similarity matrix for each representation model and directly compute the new similarity matrix using the weighted sum. Then, we apply a general threshold. Using the same example as before, the overall pairwise similarity is 0.50*0.74 + 0.35*0.86 + 0.15*0.24 = 0.707. The difference in this method is that we will compute the optimal threshold after the aggregation of the similarity matrices, and not before. Thus, for this configuration only, a validation phase is needed and is described in section 4.4. More formally, for each ga_{ij} of GS, the general similarity matrix:

$$fga_{ij} = \begin{cases} 0, & \text{if } ga_{ij} < GT \\ ga_{ij}, & \text{otherwise} \end{cases} \tag{4.7}$$

Now that we know how the similarity matrices are aggregated, we present the different content representation configurations we propose

4.4.3 Models configurations

We study different configurations:

Lexical and Semantic: it is a combination of IDF, the lexical representation model, and Universal Sentence Encoder (USE), the semantic representation model. In the rest of this paper, we will refer to this combination as **LS**.

Lexical, Semantic, and Twitter-specific: We take the **LS** combination and add specific features, namely hashtags and URLs shared in the documents. We will refer to this combination as **LSTS** in the rest of this paper.

Each of these configurations will be evaluated using the different methods presented in Section 4.4.2.

4.5 Experimentations, Results & Analysis of the combinations

4.5.1 Phases of the experiments

In this section, we present the objectives of each phase of the experiments: the Training Phase, the Validation Phase, and the Testing Phase. As we stated before, the validation phase is needed only for the GA aggregation. This process is illustrated in figure 4.13.

Training phase: The objectives of this phase are twofold: first, we want to determine the optimal threshold for each representation model. The optimal threshold is defined as the threshold value maximizing the sum of the evaluation metrics. These metrics will be presented in the next section. The second objective is to determine the relative weight of each representation model when we combine them. To do so, for each configuration (i.e. LS and LSTS), we determine the relative importance of each model according to its performances. In practice, we compute the total of the sum of the evaluation metrics for each representation model, and weight each representation model according to its contribution to this sum, as in (Becker et al., 2010). For example, suppose we take the model LS, composed of USE and IDF. The total sum of the evaluation metrics for the two models is 3.10 and the sum of the evaluation metrics of USE is 1.80. The relative weight of USE will be $1.80/3.10 = 0.58$. Respectively, the weight of IDF is 0.42.

Validation phase: This phase concerns only the General Aggregation (GA) aggregation method. The objective of this phase is to compute the optimal threshold for each configuration. The representation models are weighted according to the results of the training phase, however as the aggregation method is defined, no threshold is applied to their similarity matrix. Only a general threshold is applied to the aggregated similarity matrix. This general threshold is the threshold that maximizes the sum of the matrics, as for the model thresholds.

Testing phase: We evaluate USE and IDF and consider these models as the baselines. We also evaluate each configuration (LS, LSTS) using each aggregation configuration (SA, BA, GA). In total, during our experiments presented hereafter, we evaluate 8 representation models.

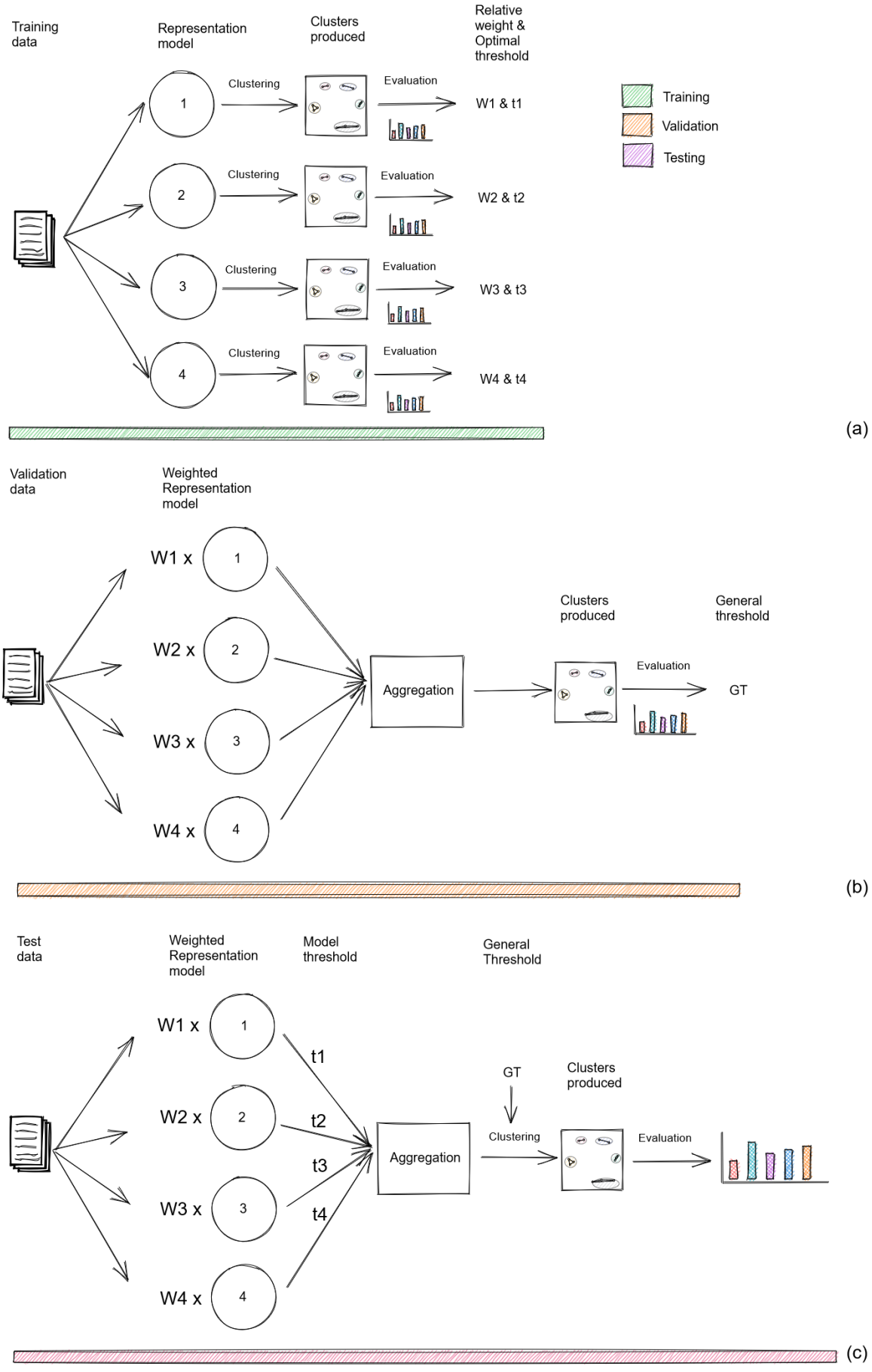

Figure 4.13: Illustration of the different phases of the process. (a) Models are evaluated to obtain their relative weight and optimal threshold; (b) A general threshold is obtained by optimizing the results; (c) Documents are represented according to the parameters obtained.

4.5.2 Experimental setup

Dataset

We use Event2012 (McMinn et al., 2013), presented in section 4.3.1. We divide the dataset into 3 equal sets: the training set, the validation set, and the testing set, each of them composed of 9 days of data.

Preprocessing

To clean the tweets, we remove from the text the user and retweet mentions and the URLs.

Evaluation methods

Each model is evaluated using B-cubed and AMI. B-cubed is presented in section 4.3.2. AMI measures how much information is shared between ground truth and the clustering assignment, adjusted to penalize random clusters. We have been inspired by the choices made in (Becker et al., 2010), where they argue that these measures "balance our desired clustering properties: maximizing the homogeneity of events within each cluster and minimizing the number of clusters that documents for each event are spread across". We are looking for the same properties, however, we decide to use AMI instead of NMI because it is rescaled such that a random clustering has a score 0, contrary to NMI.

4.5.3 Results

Training phase

The results of the training phase are illustrated in Figure 4.14. For each of the representation models (USE, IDF, Hashtags, URLs), we compute the threshold maximizing the AMI + F1 score, as it is done in (Becker et al., 2010). In Figure 4.14, the optimal threshold obtained for USE is 0.40 while the optimal one for IDF is 0.10. As we can see in the figure, USE has better performances than IDF, thus it will have a higher weight in the combinations. The obtained weights are presented in Table 4.7.

Validation phase

Like the training phase, we calculate the thresholds for LS and LSTS for the GA aggregation. The thresholds for the GA aggregation are **0.35** for LS and **0.30** for LSTS..

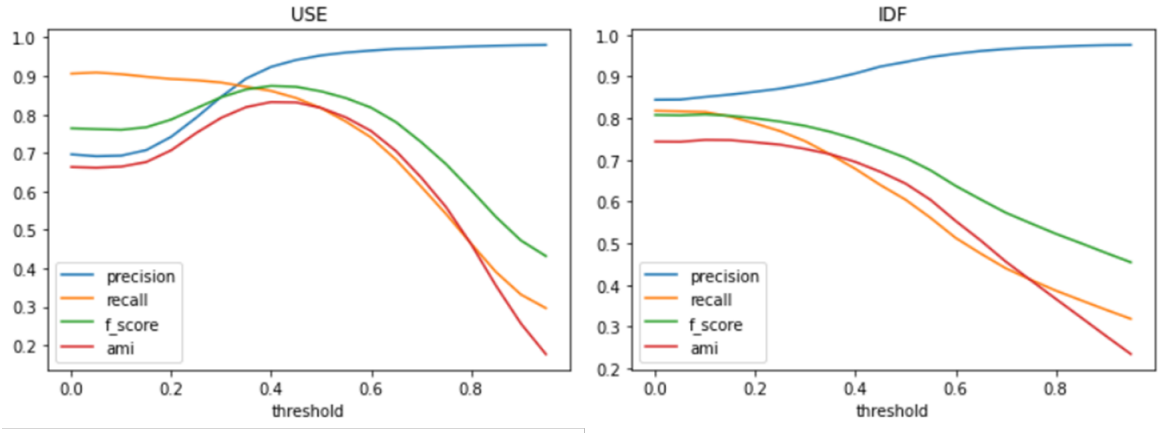

Figure 4.14: Illustration of the influence of the threshold parameter during the training phase. The objective is to learn the optimal threshold for each individual representation model. We can also see that the sum AMI + F1-Score is higher for USE than IDF, meaning its weight will be higher in the combinations.

Table 4.7: Weights and threshold for each method

Model	Threshold	Weight LS	Weight LSTS
USE	0.40	0.54	0.41
IDF	0.10	0.46	0.38
URLS	0.15	-	0.11
Hashtags	0.85	-	0.10

Testing phase

The results of the testing phase are summarized in Table 4.8. In terms of precision, LSTS/GA is performing better, with LS/GA and USE having similar performances. The BA aggregation methods seem to favor the performances in terms of recall. Finally, USE is the best performing representation model overall, achieving the best F1-score and AMI. Overall, the performances of all the models are really close, with IDF slightly lagging behind.

4.5.4 Analysis

First of all, it is interesting to note that in our context Transformer-based language model (USE) achieves the best performances, contrary to what was previously stated in the literature. We believe that the dense representations produced by Transformer-based language models are less adapted to dynamic clustering tasks like the First Story Detection algorithm, as they tend to create clusters that are more fragmented than in the classical clustering task. Methods like IDF producing sparse vectors are less sensitive to that because they already tend to produce more fragmented clusters due to their nature. Secondly, the combination methods can be interesting depending on what one is looking for in his application. To favor Precision, a combination based using the

Table 4.8: Results from the Testing Phase. The thresholds for the GA aggregation are **0.35** for LS and **0.30** for LSTS. As a reminder, GA is General Aggregation, BA is Binary Aggregation, and SA is Similarity Aggregation.

Model	Aggregation method	Precision	Recall	F1-score	AMI
USE	-	0.91 ±0.08	0.83 ±0.19	**0.85** ±0.12	**0.76** ±0.22
IDF	-	0.85 ±0.09	0.77 ±0.20	0.79 ±0.13	0.65 ±0.24
LS	GA	0.91 ±0.07	0.80 ±0.20	0.83 ±0.12	0.73 ±0.23
	SA	0.86 ±0.09	0.82 ±0.20	0.82 ±0.13	0.70 ±0.24
	BA	0.80 ±0.11	**0.86** ±0.19	0.81 ±0.12	0.69 ±0.24
LSTS	GA	**0.92** ±0.07	0.78 ±0.21	0.83 ±0.13	0.73 ±0.24
	SA	0.86 ±0.09	0.82 ±0.21	0.82 ±0.13	0.71 ±0.23
	BA	0.79 ±0.12	**0.86** ±0.19	0.81 ±0.11	0.68 ±0.23

GA aggregation method could be interesting even if USE has similar performances. It could be interesting in domains where False positives have a huge impact such as Stock market analysis where the veracity of information cannot be easily checked by a human. To favor recall, a combination using the BA aggregation method is interesting. It is interesting for domains such as Emergency planning where missing an important event could be disastrous. Overall, USE seems to be the better compromise, achieving decent performances in all the metrics and achieving better time performances than the combination, due to the necessity to only compute one similarity matrix.

4.6 Conclusion

In this chapter, we tackle the issues of document representation and clustering related to Phase 2 of our Event Detection Framework. We proposed an event detection system (EDS) and compared it to FSD, the most classical document-pivot event detection method algorithm in the literature. We showed that our system outperforms the FSD algorithm in multiple configurations.

Then, we compared different text representation models in the context of EDS. We showed that Transformer-based language models can achieve competitive results with TF-IDF, contrary to what was previously stated in the literature. We also showed the interest of fine-tuning the models in our context, which is an interesting result considering that a lot of concept drift is happening in our context.

Finally, we proposed a new combination of document representation models, based on lexical, semantic, and specific features. We showed that depending on the application, some of these combinations can be interesting, depending on if one wants to focus the performances on high precision, high recall, or overall good performances.

In the rest of this work, we want to extend our algorithm to a more realistic context of application. We focused on an annotated dataset, where all the documents are event-

related. This is not the case in most real-world applications, such as event detection on social networks. Indeed, they are full of spam and mundane conversation. Thus, in the next chapter, we investigate a more complete version of the event detection system, implementing all the modules of the event detection framework, with a special focus on the "Event detection" module and the "Event summarization and Tracking" module of the framework.

Chapter 5

Event identification : detection, summarization & tracking

In this chapter, we focus on the issues related to Phase 3 of the event detection framework. Specifically, we deal with issues related to the event detection module and the event summarization and tracking module. In the previous chapter, we introduced a first version of EDS, focused on the clustering part, and evaluated its performances on event-related documents. In a more realistic scenario, an event detection phase is needed. Indeed, most of the documents posted online do not discuss an event, some are spams that can be filtered during preprocessing steps, and some are mundane conversations that cannot be filtered this way. In this chapter, we deal with all the issues related to these steps, namely how to determine whether a cluster deals with an event, how to evaluate an event detection system when most of the documents are unlabeled, how to compare if different clusters deal with the same event and how to present the content of an event to a human in a meaningful manner.

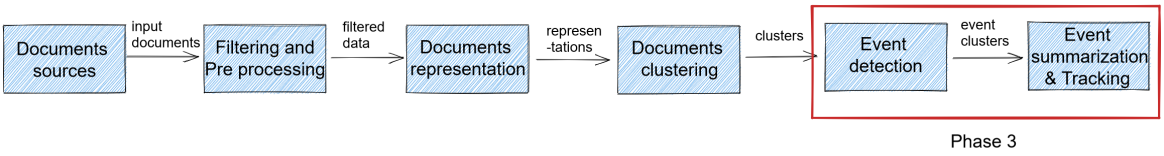

Figure 5.1: Phase 3 of the framework : Event identification

5.1 Introduction

Working on annotated datasets in a context such as Twitter without including spam or non event-related documents induces a bias in the task. As we presented earlier in this thesis, tweets are mostly spam and must be filtered and analyzed to extract meaningful information. To do so, the simple version of EDS presented in the previous chapter is not sufficient because it includes no filtering step and no analysis of the produced clusters. In this chapter, we consider the whole framework presented in Chapter 3 and present the full version of EDS. There are two main differences from the first version. First, a filtering step is performed during phase 1, to filter out spam and uninformative messages. Secondly, phase 3 is performed after phase 2. The objective of this phase is to analyze the clusters produced in phase 2, to detect which refer to an event, and then track and summarize these events.

Adding these steps allows us to apply EDS to data representative of the real world and of a real social network stream of text data. However, evaluating such an event detection system in a real-world context is not an easy task. In classical datasets such as Events2012 (McMinn et al., 2013) or Events2018 (Mazoyer et al., 2020a), a lot of the unlabeled tweets refer to labeled events, making it impossible to rely only on the labels or on the ability of the system to separate labeled from unlabeled tweets. Moreover, these datasets are not stable over time because they comply with the Twitter policy (Hasan et al., 2019) to protect user confidentiality, which means only the tweet ids are shared and the tweets must be retrieved using the Twitter API, making some tweets unavailable due to their deletion or a change in the publisher's account settings. Due to all these constraints, there is no standard benchmark dataset (Saeed et al., 2019b) and researchers usually have to rely on human annotation in addition to the original labels of the datasets, limiting the interest of old public datasets compared to new private ones. It makes their experiments non-reproducible, potentially biased, non-automated, and costly.

In this chapter, we are interested in applying our full event detection system to a dataset representative of the real world and evaluating it. To do so, we need an evaluation process that is adapted to our context and that is reproducible. Thus, in this chapter, we present different propositions. First, we introduce the two last modules of EDS to obtain the full version, built on top of the whole event detection framework. Then, we are interested in evaluating EDS on a dataset representative of the real world. To this end, we first review the different evaluation methodologies of the literature and note the lack of reproducibility of most methods. Thus, to make the evaluation reproducible and limits the necessity of human annotation, we introduce a reproducible way to compute classical metrics that suits existing datasets. This evaluation process is based on the comparison of the content of the detected events and the ground truth events. Only one parameter needs to be tuned on a small part

of the dataset to evaluate the system on the whole dataset. Sharing this parameter allows reproducing the results. The process can be used to evaluate event detection systems on classical datasets and takes into account the fact that the available tweets evolve over time by evaluating the event detection systems based on the available content. Our experiments show these evaluation measures are coherent with human evaluation. Then, we use this evaluation process to evaluate EDS and specifically the event detection and event tracking steps. Finally, we compare the performance of EDS to other systems of the literature and show that our system is competitive with these approaches.

This chapter is organized as follows: in a first section, we present the last two modules of EDS, to obtain the full version. Then, we introduce our evaluation process, how to apply it and how we evaluate its performances. Then we apply it to the last modules of EDS and compare its performance with other models of the literature. The work presented in this chapter is to be published and is currently undergoing the review process. The code relative to this chapter is available online [1].

5.2 EDS: event detection, tracking & summarization

In this section, we introduce the last two modules of the event detection system built on top of the full event detection framework, to obtain the full version of EDS.

5.2.1 General description of the process

Algorithm 3 describes the process detailed hereafter. We take the process where it ended in Section 4.2.2, namely after we obtained the clusters of documents. After this step, we perform the Event Detection phase, to evaluate which cluster refers to an event, to obtain a set of detected events annotated $DE^k \subseteq W^k$, k referring to the number of the window. The clusters are evaluated using different metrics defined in section 5.2.2. Then, all the Detected Events DE^k of a window k are compared to the detected events of the previous window DE^{k-1} during the event tracking phase. Chains of events are created to determine when an event starts, how long it lasts and when it finishes. This process is described in section 5.2.3. Finally, during the Event Summarization phase, each detected event $DE^k_j \in DE^k$ is represented in a way that is understandable for a human being.

5.2.2 Event detection

To evaluate whether a cluster deals with an event, we use two classical measures from the literature. It is a classical combination also used in other event detection systems

[1] https://gitlab.com/Emaitre/eventdetectionsystem

Algorithm 3: EDS

input : window time W, a stream of tweets S
output: for each window t : a list of clusters, An updated list of event chains

```
// This is a background task creating the windows
while Timer < WindowTime do
    foreach tweet in S do
        if filter(tweet) then
            FilteredTweets ← tweet ;
        else
            Discard(tweet);
        end
    end
end
// This can be run in parallel
// For each window t
ListClusters_t ← EDSClusteringPart(FilteredTweets);
foreach (c_t in ListClusters_t) do
    if EventDetection(c_t) then
        ListEventClusters_t ← c_t;
    else
        Discard(c_t);
    end
end
Links ← Linking(ListEventClusters_{t-1}, ListEventClusters_t);
foreach (c_t in ListEventClusters_t) do
    if ((c_{t-1}, c_t) in Links) then
        Update(ListChains(c_{t-1}), c_t);
    else
        NewChain(ListChains(c_t));
    end
end
Return ListClusters_t, ListChains
```

(Hasan et al., 2019).

First, we use cluster entropy defined by (Petrović et al., 2010) :

$$H_{cluster}(C_j^k) = -\sum_w \frac{n_w}{N} log \frac{n_w}{N},$$

where n_w is the number of times word w appears in the cluster C_j^k, and $N = \sum_w n_w$ is the total number of words in the cluster. Following the results presented in (Petrović et al., 2010), we discard the clusters with low entropy (≤ 3.5) as non event clusters. Secondly, we compute user diversity (Kumar et al., 2014):

$$U_{cluster}(C_j^k) = -\sum_u \frac{n_u}{N_t} log \frac{n_u}{N_t}$$

where u is a user who posted a tweet in the cluster C_j^k, n_u is the number of tweets published by user u which are part of the cluster C_j^k, and $N_t = |C_j^k|$ is the total number of tweets in the cluster. We discard clusters with low user diversity (≤ 0). The entropy threshold ensures that a minimum amount of information is contained in a cluster. A positive user diversity value ensures that a cluster contains tweets from more than one user.

5.2.3 Event tracking & summarization

This module is really important for real-world applications as events evolve through time and their potential consequences might be different as they unfold. It is also important to provide a meaningful representation of the events that can be understood by humans.

Because of our choice to discretize the stream using independent windows, it is necessary to establish some links between the events detected in each window since the events are not stopping at the end of a window. A solution proposed in (Fedoryszak et al., 2019) is to create events chains. A similarity measure is computed between detected events and if the similarity is above a fixed threshold, the detected events are considered as dealing with the same event and a link is created between them. The authors apply this method to a feature pivot approach, hence their clusters are constituted of features and they directly compare the features present in each cluster of the pair. We investigate how to link clusters constituted of documents.

Concerning the summarization of a group of documents to obtain a meaningful representation, it is a research area in itself. We did not study in detail this part and did not conduct any meaningful research. We consider that we will provide some of the tweets of the clusters and the most represented named entities to the users to let them understand the content of each cluster. Providing better summaries is left for future work.

Presentation of the tracking method

After the event detection phase, we obtain a set of detected events per window. The idea is to compare the events of window W^n to the events of window W^{n-1}. To do so, we build a bipartite graph where the events of window n are on the right-hand side of the bipartite graph and the events of window $n-1$ on the left-hand side of the bipartite graph. Each event is considered as a vertex and an edge is drawn between each pair of events if their similarity is above a defined threshold. We reproduce this process for each pair of windows to obtain cluster chains. We then analyze these chains to determine when an event starts, when it ends, and its duration. An example of such a chain is provided in figure 5.2, which illustrates the chaining of the ground truth events of Events2012 and thus what our chaining should look like when optimized.

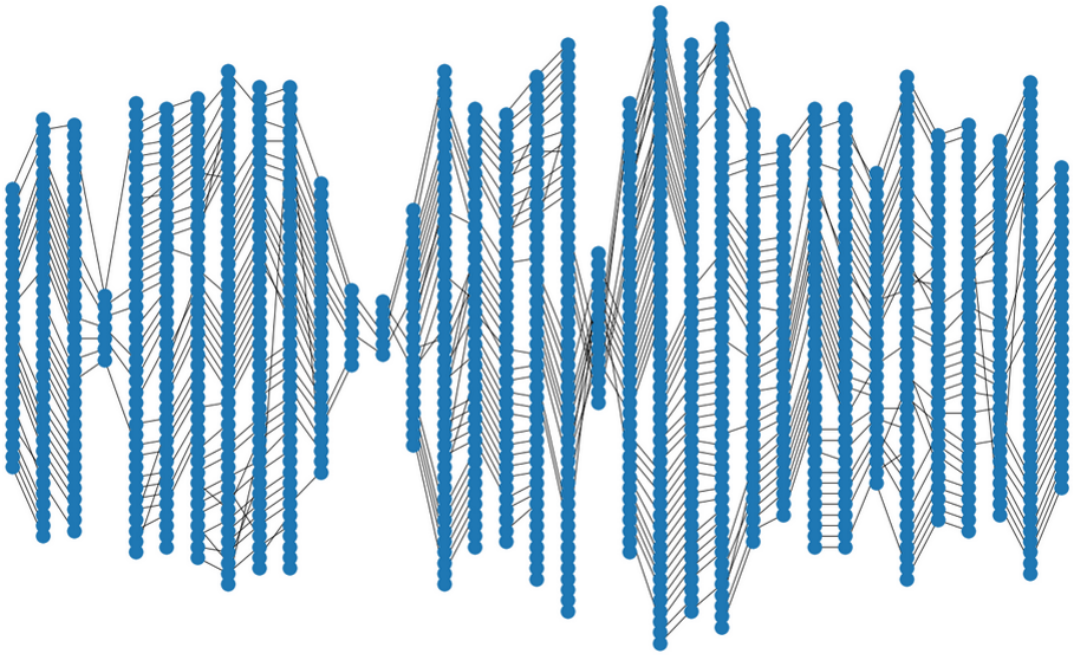

Figure 5.2: An example of a cluster chain. Each column is a window, each dot is an event. This is the ground truth chain from the Event2012 dataset. For the sake of visualization, we used windows of $\tau = 2000$ tweets.

Event cluster representation

We consider different methods to measure the similarity between two events:

- **Common entities -** This method measures the common entities between two events. We extract all the named entities of each detected event and construct a Term-Frequency vector of the entities. We then compute a cosine similarity between each TF vector.
- **Representative document -** We compute the document which has the highest overall similarity with all the other documents of the event cluster. Then we

measure the similarity of this document with the representative document of each other event cluster using a document representation and a similarity measure, just as we did when we computed the clusters in earlier phases.

- **Central document -** We compute which document has the highest number of neighbors in the event cluster. We then compare this document to other central documents as presented before.

Now that we presented the different choices we made for each module of the event detection framework, we divide the rest of this chapter into three sections. In a first section, we present a new process to evaluate event detection systems, for the reason explained in the introduction, section 5.1 and the different experiments we conducted to validate the evaluation process. Then, in a second section, we evaluate the event detection system using this evaluation process. Finally, in a third section, we compare the performances of EDS to other systems of the literature.

5.3 A new evaluation process based on content similarity

In this section, we present our evaluation process based on content similarity, justify its necessity and evaluate its performances. The section is organized as follows. First, we present some related work on how different event detection systems are evaluated. Then we describe in detail our evaluation process. Finally, we present the different experiments we conducted to validate the process and discuss our results.

5.3.1 Related work on event detection models evaluation

The approaches presented in section 2.4 are either evaluated on public or private datasets. Only a few datasets for event detection on social media are publicly available, because of the cost of the annotation process, the difficulty of correctly covering events in gold standards (McMinn and Jose, 2015), (Soni and Pal, 2017) and the difficulty to create datasets that are not biased toward high impact tweets (Mazoyer et al., 2018). There is no standard benchmark dataset (Saeed et al., 2019b), thus, a lot of event detection systems are evaluated on private datasets (Petrović et al., 2010), (Li et al., 2012), (Becker et al., 2010), (Boom et al., 2016), (Fedoryszak et al., 2019) which makes the results non-reproducible. In terms of public datasets, Events2012 (McMinn et al., 2013), presented in detail in section 4.3.1, is the most used and is adapted to the open-domain detection task. 150k labeled tweets are provided but most of the tweets (120M) are unlabeled. Some of these unlabeled tweets refer to ground-truth events, some to unlabeled events, and some are spams. Hence, one cannot evaluate the performances of its event detection system based on its ability to separate annotated, event-related tweets from unlabeled tweets.

Thus, different strategies are implemented to evaluate event detection systems on this dataset. Some authors evaluate them only on the labeled portion of the dataset (Mazoyer et al., 2020a), assuming that all tweets are event-related, which is commonly done in the literature (Becker et al., 2010), (Boom et al., 2016) and similar to what we did in chapter 4. This hypothesis is convenient to evaluate the clustering models but is not representative of what happens in a real-world context where not all the documents are event-related. When working with the whole dataset, evaluation methods rely on human annotation in addition to the labels (Hasan et al., 2019), (Morabia et al., 2019), (Pandya et al., 2020), making the results difficult to reproduce. Only two papers use automated evaluations on Events2012. In (McMinn and Jose, 2015), the authors use both automatic and human annotations. They rely on the labeled tweets of the cluster and not on the unlabeled ones, disregarding the majority of the content of each cluster. In (Edouard et al., 2017) the authors argue that "since we include both event-related and not event-related tweets, we consider an event as correct if 80% of the tweets belong to the same event in the ground truth". Considering that only 0.2% of the tweets are labeled and a lot of unlabeled tweets refer to ground-truth events, it is a surprising result because separating labeled from unlabeled tweets is not a desirable objective. Events2018 is similar to Events2012 but in French and is less commonly exploited in the literature.

Other datasets, such as the FA Cup/Super Tuesday/US Election datasets (Aiello et al., 2013) or more recent like the English Premier League datasets and Brexit Super Saturday datasets (Hettiarachchi et al., 2021), are also widely used in the literature. In these datasets, the tweets are not annotated but some important keywords depicting the events are provided and event detection systems are evaluated on their ability to retrieve these keywords from the documents. The evaluation on these datasets can be automated but the events are not overlapping with each other, which is not representative of a real-world context where different events are happening at the same time.

Thus, there is no reproducible method to evaluate event detection systems on publicly available datasets representative of the real world. We summarize this discussion in table 5.1 In the rest of this section, we define a new way to compute classical evaluation metrics by combining the keyword approach introduced in (Aiello et al., 2013) and datasets like Events2012 (McMinn et al., 2013) which satisfies these constraints. Using this evaluation process, tuning a single parameter on a small annotated part of the dataset is enough to evaluate an event detection system on the full dataset.

Table 5.1: Comparison of how different event detection approaches from the literature are evaluated.

Article	Dataset	Manual annotation	Evaluation metrics	Clustering	Approach type
(Li et al., 2012)	Private	Yes	No of events, Precision, Recall, DER	Yes	Feature pivot
(Morabia et al., 2019)	A portion of Event2012	Yes	No of events, Precision, DER	Yes	Feature pivot
(Pandya et al., 2020)	Event2012	Yes	Precision, Recall, DER	Yes	Feature pivot
(Edouard et al., 2017)	Event2012	No	Precision, Recall, F1 score	No	Feature pivot
(Fedoryszak et al., 2019)	Private	No	detected/merged/ duplicate event fraction, Consolidation, Discrimination, Clustering score	Yes	Feature pivot
(Asgari-Chenaghlu et al., 2020)	FA cup, Super Tuesday, US elections	No	Top-K topic-recall, Top-2 keyword-precision	No	Feature-pivot
(Hettiarachchi et al., 2021)	English Premier League, Brexit Super Saturday	No	Precision, Recall, F1 score	Yes	Feature pivot
(Petrović et al., 2010)	Private	No	Average precision on gold standard	Yes	Document pivot
(McMinn and Jose, 2015)	Event2012	Both	Precision, Recall, F1	Yes	Document Pivot
(Naaman et al., 2011)	Private	No	F1 score	Yes	Document pivot
(Boom et al., 2016)	Private	No	Split Error, JS divergence	Yes	Document pivot
(Hasan et al., 2019)	A portion of Event2012	Yes	Precision, Recall	Yes	Document pivot

5.3.2 A novel evaluation process

General description of the task

As stated in section 2.4, the output of event detection systems is a set of detected events. The most important question is how to decide whether a detected event $C_j \in DE$ (with DE the set of detected events) matches a ground truth event $e_i \in GTE$ (with GTE the set of ground truth events) to compute evaluation metrics. The objective of our evaluation process is to automate this decision to reduce the need for human annotation. The process applies to any event detection system that produces clusters of documents or features and uses them to detect events, under the condition of being able to find the original document of a feature, to associate it with its label, which is usual Edouard et al. (2017), Hasan et al. (2019), Morabia et al. (2019).

Matching ground truth and detected events

Figure 5.3: Illustration of the matching procedure. For each detected event containing at least a labeled document, an association is created with a ground truth event. This ground truth event corresponds to the majority label of the labeled documents in the detected event. Then, if their respective named entities are similar, the association becomes a matching.

A DE event can contain both labeled and unlabeled documents. However, these unlabeled documents can refer to a GT event e_i. So, we need to evaluate event detection systems based on their ability to separate documents assigned to different GT events but also analyze unlabeled documents they are grouped with. We propose a new method to calculate a matching between $C_j \in DE$ and $e_i \in GTE$, by evaluating the similarity between C_j and e_i. It is divided into two parts, association, and matching. Figure 5.3 illustrates the proposed method.

Association - We create an association between C_j and e_i if there is at least one labeled document in C_j and the most frequent label k appearing in C_j is the label i of e_i. We name this set ADE. This eases the human annotation phase, described in section 5.3.2. This phase is not sufficient to check if an event cluster truly corresponds to an event. So, we add a second step named matching.

Matching - We check whether e_i and C_j are similar enough. In terms of similarity, different types of cluster representation can be considered, i.e. lexical similarity, se-

mantic similarity, or based on features such as hashtags or images. If a detected event C_j satisfies both association and matching conditions, we consider there is a match between C_j and e_i. We name this set MDE.

Implementation and illustration of the added value of the proposed process

In our case, we decided to check whether the named entities of e_i are similar to the named entities of C_j. We compute the cosine similarity between the term-frequency vectors of named entities of C_j and e_i. If the similarity is above a defined threshold, we consider that they refer to the same event. We chose the cosine similarity because it is the most common similarity measure in NLP Aggarwal and Zhai (2012).

The only parameter of the method is the threshold t applied to the cosine similarity. This threshold is obtained during a short training phase, described in section 5.3.2, implying a few human annotations. Once this threshold is obtained, the rest of the associations can be automatically evaluated.

One may wonder why we chose to evaluate a match between a GT event and a DE event based on named entities and what is the added value of the method compared to the usage of the label of the tweets. First, we decided to choose this because of how an event is defined. As a reminder, we use the following definition: **An event is a significant thing that happens at some specific time and place. Something is significant if it may be discussed in the media. For example, you may read a news article or watch a news report about it. It is identified by a group of entities (e.g. people; location) that is discussed in the documents dealing with the event. The eventful conversation can change over time, and our data model for an event should reflect this.** Thus an event is characterized by a group of entities that is discussed in the documents. We consider that if the entities discussed in the DE event are similar enough to those discussed in the associated GT event, then there is a match. Second, we illustrate the added value of the evaluation method in figure 5.4. As we can see, in both cases the DE event is associated with the same GT event, meaning that at least one tweet of the DE event refers to this GT event, and the majority of the labeled tweets refer to this event. In the first case, the DE event discusses the GT event, as we can see from the sample tweets and the named entities. Using the evaluation method, the association becomes a match. In the second case, the sample tweets discuss different topics and the named entities reflect this as well. The match between this DE event and the GT event will not be confirmed due to the dissimilarity between the DE named entities and the GT named entities. This type of evaluation would not be possible using the labels only and without relying on the content of the clusters.

Description of the associated event	A court in Moscow, Russia, frees one of the three Pussy Riot members at an appeal hearing.
Tweets from GT	'Pussy Riot member freed after Moscow court appeal', 'BBC News: Pussy Riot appeal opens in Moscow: A Moscow court begins an appeal hearing for three activists from pu...', 'Moscow court frees one jailed Pussy Riot member'
NE From GT	('Pussy Riot', 117), ('Moscow', 71), ('Russian', 33), ('One', 27), ('1', 16)
Sample Tweets from DE	'One Pussy Riot member freed on appeal by Russian court', 'Pussy Riot member freed after Moscow court appeal', 'One Pussy Riot Member Freed by Moscow Court Two year sentences for two other defendants are upheld by the court', 'Moscow appeals court frees Pussy Riot member Yekaterina Samutsevich but upholds sentences for Nadezhda Tolokonni', 'One Pussy Riot member freed on appeal by Russian court'
NE from DE	('Pussy Riot', 43), ('Moscow', 18), ('Russian', 17), ('One', 13), ('two', 13), ('third', 7), ('1', 7), ('3', 6), ('MOSCOW Reuters', 5), ('three', 4)

Figure 5.4: An example of an association that will turn to a match using the evaluation method. The detected event clearly discusses the associated ground truth event.

Description of the associated event	A court in Moscow, Russia, frees one of the three Pussy Riot members at an appeal hearing.
Samples Tweets from GT	'Pussy Riot member freed after Moscow court appeal', 'BBC News: Pussy Riot appeal opens in Moscow: A Moscow court begins an appeal hearing for three activists from pu...', 'Moscow court frees one jailed Pussy Riot member'
NE From GT	('Pussy Riot', 117), ('Moscow', 71), ('Russian', 33), ('One', 27), ('1', 16)
Sample Tweets from DE	'Australia s PM Slams Misogynist Opposition Leader Gillard Vs Abbott via auspol Tonyabbott', 'Watching these puppies play since I made her lunch for today', 'Pussy Riot One defendant freed A Moscow court frees one of the convicted women from the punk band Pussy Riot', 'Those 5 seconds you have to wait on YouTube are the longest 5 second of your life', 'Regardless of everything I m in a pretty good mood today'
NE from DE	[('today', 38), ('Pussy Riot', 16), ('3', 9), ('this morning', 9), ('Yay', 8), ('2', 8), ('Moscow', 8), ('Don', 7), ('morning', 6), ('2 hours', 6)]

Figure 5.5: An example of an association that will not turn to a match using the evaluation method. The detected event is composed of multiple topics.

Evaluation metrics

Once the matches between detected events and ground truth events are established, we compute the following metrics to evaluate the systems:

$$Precision = \frac{|MDE|}{|ADE|},$$

where $|MDE|$ is the number of detected events that match with a ground-truth event and $|ADE|$ is the number of associated detected events, i.e. that contain at least one labeled document.

$$Recall = \frac{|MGTE|}{|GTE|},$$

where $|MGTE|$ is the number of ground truth events that match a detected event and $|GTE|$ is the number of ground truth events.

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall},$$

These metrics do not take into account the events that are detected multiple times. Thus, we also use the **Duplicate Event Rate (DER)** introduced in (Li et al., 2012), which corresponds to the percentage of events that are detected multiple times. We also introduce a new evaluation method because recent papers lack a measure of time precision (Edouard et al., 2017), (Morabia et al., 2019), (Pandya et al., 2020), an important parameter for an efficient event detection system. It is called **Time Precision Rate (TPR)**. It evaluates the ability of an event detection system to detect events as early as possible. It is defined as the ratio between the number of ground truth events detected in time and the total number of detected events. We consider a ground truth event as detected in time if the window in which it is detected in the same window in which its first labeled tweet is published. To evaluate the accuracy of the evaluation method, TPR will not be considered, but we will use it to evaluate our event detection system in the next section.

Application procedure

Algorithm 4 illustrates the application procedure. First, the event detection system is applied to the dataset to obtain the detected events, i.e. clusters of documents. In line 2, all the detected events containing at least one annotated document are retrieved and associated with the ground truth events corresponding to the majority label, to obtain ADE. In line 3, a portion of ADE is annotated to obtain the ground truth matches, used to calibrate the threshold for the cosine similarity. The number of manual annotations can vary but we recommend annotating at least a few tens of associations. Using these annotated associations, one can compute the ground truth evaluation metrics and use them as objectives values during the training phase of line 4. During the training phase, the evaluation method is applied to all the annotated associations, using all possible threshold values. The optimal threshold retained is the one that minimizes the mean squared error between the ground truth evaluation metrics and the evaluation metrics obtained using the evaluation method. This optimal threshold can then be applied to the rest of the associations to estimate the performances of the event detection system, which corresponds to lines 5 to 12.

In the next section, we present the experiments we conducted to validate the performance of our evaluation process.

Algorithm 4: Application procedure of the evaluation method

input : Event detection system EDS, Dataset D
output: An evaluation of the method

$DetectedEvents \leftarrow EDS(Documents)$;
$ADE \leftarrow GetAssociations(DetectedEvents)$;
$trainset \leftarrow Annotation(ADE)$;
$t \leftarrow TrainThreshold(trainset)$;
foreach *Event E in ADE* **do**
    $NEC \leftarrow GetNE(C)$;
    $NEGTE \leftarrow GetNE(Ei)$;
    if *Cosine(NEC, NEGTE)* $> t$ **then**
        $MDE \leftarrow (C, i)$;
    end
end
$Evaluation(MDE)$

5.3.3 Assessment of the evaluation process

The objective of this section is to evaluate the performance of the evaluation process, particularly the matching method. We evaluate its sensitivity to the size of the training set, to the event detection system, and we compare the results obtained using the method to the results obtained using human annotation.

Experimental configuration

We use the following configuration :

Dataset - We experiment on the Event 2012 dataset (McMinn et al., 2013). We considered the 57M tweets we retrieved, and applied different filtering rules to these tweets: removing the tweets without named entities as suggested in (McMinn and Jose, 2015), removing retweets, and cleaning tweets to remove URLs, user mentions, hashtags. We obtain a set of 16M tweets.

Event detection systems - We use two variations of EDS presented earlier in this chapter, one using TF-IDF (System 1) and the other using USE (System 2) as a text representation model. We do not consider the "Event Tracking & Summarization" module in this experiment.

Named Entity Recognition - We used Spacy[2] (Honnibal et al., 2020) as NER. We used the model "en_core_web_sm"[3] for all our experiments.

Experiments

In this section, we evaluate the capacity of our evaluation method to obtain results similar to human annotators and the sensitivity of the threshold depending on the

[2] `https://spacy.io/`
[3] `https://spacy.io/models/en`

context. We annotate a part of the dataset as described in section 5.3.2, and use a cross-validation approach in these experiments. In particular, we annotated 48 annotated hours of the dataset and then split it into 6 8-hour folds. To show the sensitivity to the training/testing ratio, we evaluated the results using different configurations : **C1** for which we use a ratio of 1 fold of training and 5 of testing, **C2** a 2/4 ratio, and **C3** a 5/1 ratio. For C2, we randomly chose 6 settings among all the possible combinations. We used the same folds for each system.

First, we describe the annotation phase.

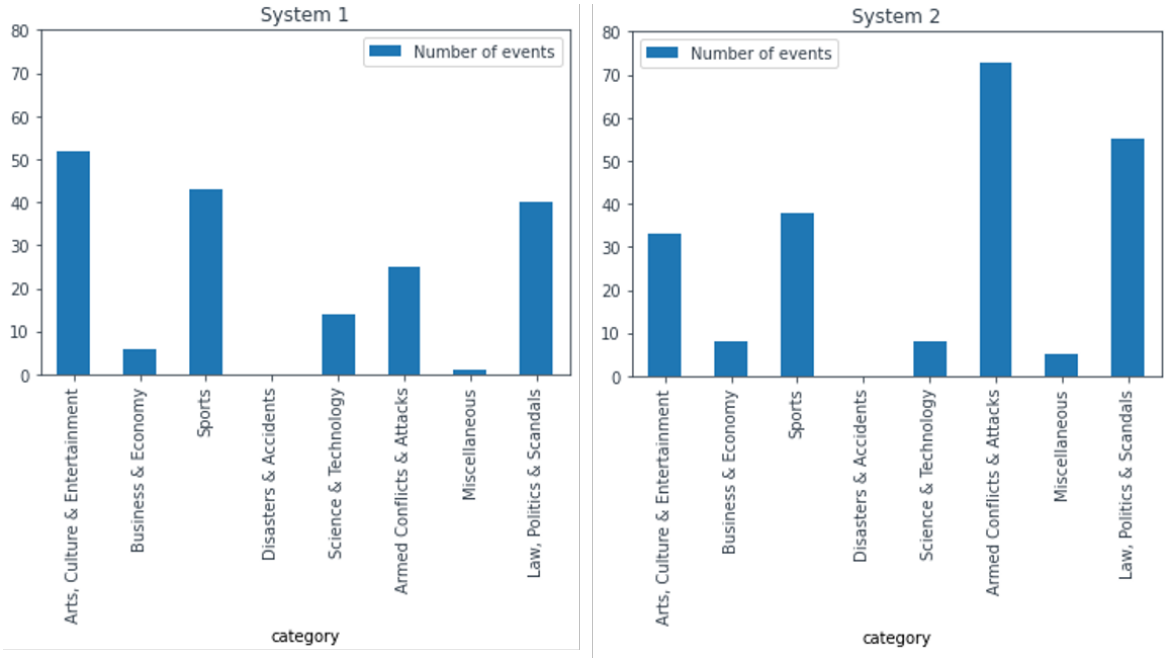

Figure 5.6: The number of events per category according to the system. This number varies according to the system because the constitution of the candidate event clusters depends on the system.

Annotation of the associations - We asked three annotators to annotate all the associations of the first 48 hours of the dataset for the two event detection systems presented hereafter. It corresponds to roughly 200 events for each system. To annotate the associations, we presented the following elements to the annotators: the description of the ground truth event given by the authors of the dataset, a set of 5 random tweets extracted from the ground truth, the 5 most frequent entities of the ground truth. We also provided 100 tweets from the detected events, selected randomly as well as the 5 most frequent entities of the detected event. The annotators were asked to annotate the association as "1" if they could easily understand if the detected event is discussing the ground-truth event using the information provided, and "0" otherwise. We calculated cohen's kappa coefficient (Landis and Koch, 1977) for each pair of annotators and for each event detection model to evaluate the agreement between them. Results are presented in table 5.2. Even if the agreement is strong (most of them > 0.60), it seems that reaching a consensus is difficult, even between human annotators. For

Figure 5.7: Agreement between annotators depending on the category for (a) system 1 and (b) system 2. As we can see, annotators have a strong agreement on some categories like Business & Economy. Other categories, such as Miscellaneous are harder to agree on. These results have to be analyzed considering the imbalance between categories, i.e. some categories like Miscellaneous have only a few events, leading to strong variations in terms of percentage of agreement.

the rest of this paper, we assign the value 0 (resp. 1) to an association if at least two annotators gave the value 0 (resp. 1). An illustration of the agreement between annotators depending on the category is presented in Figure 5.7.

Threshold sensitivity - We evaluate the error according to the threshold value for both systems and the size of the training needed to find the optimal threshold. We apply the Application Procedure described in section 5.3.2 for each system to obtain in the cross-validation setting presented earlier.

Similarity to human annotation - We compare the differences between evaluation

Table 5.2: cohen's kappa coefficient for each configuration

	Annot 1 / Annot 2	Annot 1 / Annot 3	Annot 2 / Annot 3
System 1	0.82	0.84	0.74
System 2	0.64	0.66	0.48

metrics (i.e. *Precision*, *Recall* and F_1) obtained using the annotations (GT) and the evaluation method (EM). We also compare the set of human annotations and the set of predictions made by the method for the annotated associations in ADE. The evaluation metrics can be equal even if the paired predictions are different. Indeed, if the number of matches is the same in the two sets but not on the same associations, the precision and recall would be equal but the two sets would not be consistent with each other. To evaluate this, we use McNemar test McNemar (1947) to evaluate the following H0 hypothesis: "the two sets are consistent with each other". We consider H0 as rejected if p-value < 0.05. This test measures the consistency in responses across two variables for paired data, which is our case since we compare the results for the same subset of ADE. In practice, the idea is to make sure that each event of ADE annotated as "1" (resp. 0) in the human annotation corresponds to a "1" (resp. "0") in the predictions of the evaluation method.

Results and discussion

In this section, we first present our results and then analyze them.

Threshold sensitivity - The results are illustrated in figure 5.8. The optimal value depends on the fold and the system. As we can see, for S2 the optimal threshold can be identified in all configuration, even if the identification is easier with a larger training set. For S1, it is not very clear in C1 which is the optimal threshold but it appears in C2. For both system, the optimal threshold can be found with only a few annotations, the error value is relatively stable around the optimal threshold value and the error is low.

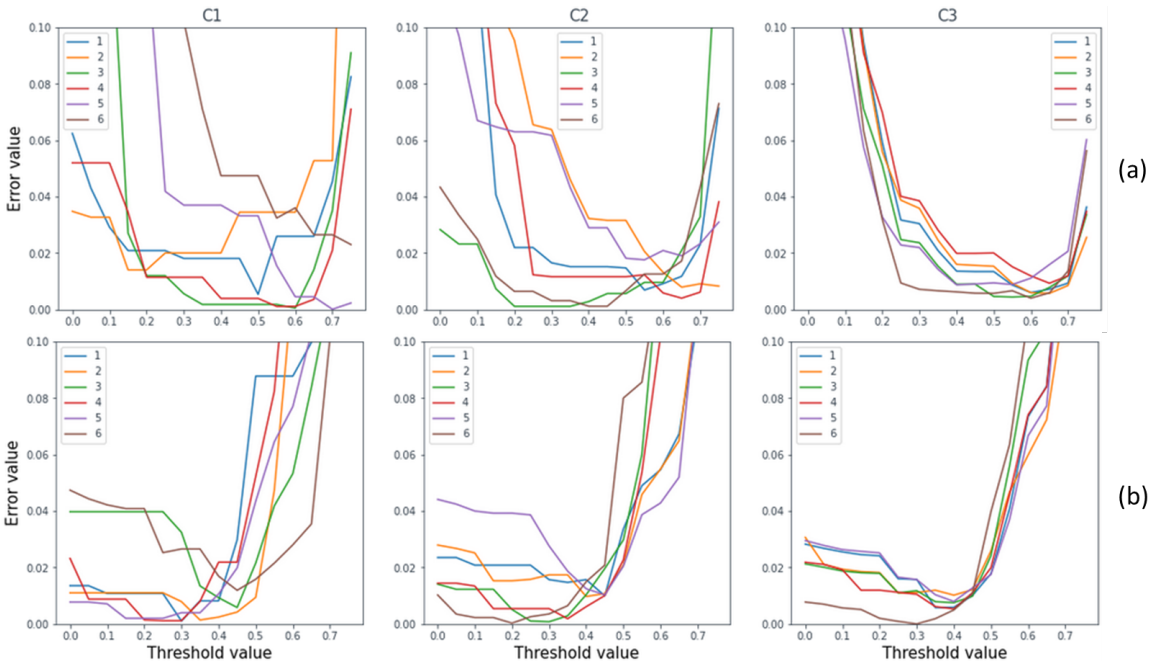

Figure 5.8: Impact of the threshold value on the error for (a) system 1 and (b) system 2. Each color corresponds to a fold.

Similarity to human annotation - Results are summarized in table 5.3. We can see that the metrics are similar, particularly for C3. For 4 of the 6 folds of C3 (folds 1, 3, 4, and 6) an interesting result is that the dissimilarities between EM and GT are the same for each model, i.e. when EM underestimates or overestimates the results, it does it for both systems. We also compare the sets of predictions and human annotations using the McNemar test. Results are reported in table 5.4. As we can see, H0 is never rejected for C3. However, concerning C1, H0 is rejected for a few fold, particularly for S1. Finally, for C2, H0 is never rejected for S2 and a few are rejected for S1, with p-value close to 0.05. Thus, it seems that annotating around 1/3 of the dataset could be enough to consistently reproduce human annotation.

Hence, the evaluation method is dependant on the system to evaluate and it seems that for S1, more annotations are required. It is coherent with the results of the threshold sensitivity section. For both systems, only a few annotation is needed to find the optimal threshold and, when tuned properly, the evaluation method is consistent with human annotation, validating its performance.

Discussion - The first element we want to discuss is the usage of Named Entity Recognition in the context of social networks. One could argue NER models in general are not mature enough to be used to evaluate systems in this context. During our experiments, we realized by manual checking that the performance is quite good and most of the important entities are retrieved. This observation might be biased due to the age of the dataset. Indeed, most of the entities are probably not novel to the model, considering the training data used for the NER we used. In the context of more recent datasets, the performance might change. However, considering the available datasets of the literature and the difficulty to create new ones, we believe this is not the most usual context. Moreover, as we use the same NER for both the ground truth events and the detected events, detected and undetected entities will be the same for both sets, so we believe the performance will not be drastically altered.

Concerning the performance of the evaluation method, we can see that for S2, using 1/6 of the training set is enough to find the optimal threshold of the annotated set and use it to evaluate the system on the full dataset, which is a good improvement compared to the literature. For S1, a few more annotations are needed, but we still found interesting results in C2. In both cases, using 1/3 or less of the annotated set as training set is enough to find the optimal threshold. Another interesting point raised is that the evaluation method seems to be coherent between systems, *i.e.* if it overestimates or underestimates the results, it does for both systems, enabling a fair comparison. Finally, the evaluation process automatically associates the DE event and the corresponding GT event, meaning that the annotation is shorter, faster and easier.

The optimal threshold obtained can be shared by researchers so anyone can reproduce the evaluations using that threshold. We believe this is a step towards re-

Table 5.3: Results for each fold. The objective is to have values as close as possible between EM and GT for each system S and configuration C. Due to the construction of the folds, values are not supposed to be similar between different configurations.

			1			2			3			4			5			6		
			P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
S1	C1	EM	.35	.51	.41	.51	.64	.57	.33	.40	.36	.33	.46	.38	.33	.39	.36	.22	.27	.24
		GT	.35	.41	.38	.36	.40	.38	.40	.40	.40	.37	.36	.36	.47	.42	.44	.41	.42	.41
	C2	EM	.32	.43	.37	.41	.45	.43	.45	.58	.51	.34	.47	.40	.34	.50	.40	.33	.50	.40
		GT	.36	.37	.36	.54	.48	.51	.33	.33	.33	.44	.44	.44	.35	.44	.39	.32	.37	.34
	C3	EM	.58	.21	.31	.41	.18	.25	.32	.18	.23	.55	.21	.30	.21	.20	.20	.34	.41	.37
		GT	.74	.23	.35	.64	.20	.30	.32	.20	.27	.55	.23	.32	.14	.11	.12	.34	.23	.27
S2	C1	EM	.82	.87	.84	.80	.84	.82	.74	.76	.75	.87	.87	.87	.89	.90	.89	.80	.79	.79
		GT	.83	.77	.80	.80	.76	.78	.84	.78	.81	.84	.79	.81	.84	.77	.80	.89	.85	.87
	C2	EM	.72	.73	.72	.86	.78	.82	.81	.83	.82	.80	.84	.82	.79	.76	.77	.82	.78	.80
		GT	.82	.73	.77	.90	.84	.87	.82	.75	.78	.83	.70	.76	.88	.82	.85	.82	.78	.80
	C3	EM	.91	.74	.82	.96	.87	.91	.96	.84	.90	.84	.71	.77	.79	.77	.78	.64	.75	.69
		GT	.94	.79	.86	.94	.83	.88	.87	.74	.80	.87	.76	.81	.85	.77	.80	.69	.62	.65

Table 5.4: p-value for the McNemar test for each fold

		1	2	3	4	5	6
S1	C1	1	1e-4	0.12	0.33	4e-3	4e-5
	C2	0.44	0.04	0.02	0.08	1	1
	C3	0.50	0.13	1	1	0.38	1
S2	C1	1	0.47	0.02	0.5	0.16	0.01
	C2	0.05	0.33	0.87	0.64	0.05	0.38
	C3	0.69	1	1	0.55	0.34	0.81

producibility. The major weakness of our evaluation method compared to human annotation is that it evaluates only detected events containing at least one labeled document, while human annotators can evaluate any detected event. Considering the wideness of the original annotated set, we think our method ensures a good approximation of the performances and that the gain in reproducibility and time is worth this trade-off.

Overall, the evaluation method is consistent with human annotation. We think this is because it is consistent with the definition of events that we used and with how a human analyzes a large number of documents: one observes the most frequent entities and compares them with the entities of the description of the event.

5.3.4 Partial conclusion

In this section, we presented an evaluation method that greatly reduces the need for human annotation in a context of a massive amount of text data. We applied this method on Twitter and it achieves great performance, is consistent with human annotation, and is a good step toward reproducibility of the results, as only one parameter needs to be shared to reproduce the results.

In the next section, we apply this evaluation method to evaluate our event detection model.

5.4 Experimentations, Results and Analysis

This section is divided into two subsections. First, we present the experiments related to the “Event Detection” module. Then, we present the experiments linked with the “Event Tracking” module. As a reminder, we did not conduct any meaningful experiments on the summarization part, which is left for future work.

5.4.1 Event detection

Dataset

We use the same configuration described in 5.3.3.

Table 5.5: Weights and threshold for each method

Model	Threshold	Weights LS
USE	0.60	0.54
IDF	0.45	0.46

Table 5.6: Threshold for the cosine similarity of the evaluation method for each text representation model

	USE	IDF	LS
Threshold	0.40	0.60	0.50

Experimental configuration

We use windows of 1 hour as explained in section 4.3.3. For the sake of comparison, we experiment using 3 text representation models: USE, TF-IDF all tweets, and LS-SA, all of them presented in Section 4. To obtain the detected events after the clustering phase, we apply the measures presented in section 5.2.2, namely entropy and user diversity.

The weights and thresholds we used are summarized in table 5.5. We chose to raise the threshold for both USE and IDF because the windows contain much more tweets in comparison with the annotated dataset. Thus, a greeter granularity is needed. We determined these thresholds empirically.

Evaluation method

We evaluate the results using Precision, Recall, F1 Score, DER and TPR calculated using the evaluation method presented in section 5.3. The thresholds used are summarized in table 5.6. We did not perform any annotation for the LS model, we chose the threshold according to the weight of each model and their relative threshold.

Results

We display different results to illustrate different aspects. The results are displayed in Table 5.7. As we can see, we can draw the same conclusions as before: USE performs the best, far above IDF. The LS combination is competitive in terms of recall and number of detected events, but the precision is lower than USE. Overall, USE is performing better in nearly every metric, except DER. Notably, DER is not necessarily a metric that we want to lower. Indeed, in the dataset, some events are really general and could be divided into subevents. Having a high DER might signal that we detect different subevents that are not labeled in the dataset.

The second aspect we want to highlight is which events are detected by each model. This is illustrated by Figure 5.9. As expected, USE has better coverage in terms of

Table 5.7: Evaluation metrics on the whole dataset.

	# of events	Precision	Recall	F1 Score	DER (%)	TPR (%)
USE	323	0.68	0.65	0.66	8	30
IDF	212	0.25	0.42	0.31	7	21
LS	305	0.53	0.61	0.53	5	25

percentage than the other models. The coverage of events is nearly similar for LS, with approximatively the same percentage of detected events as USE. Interestingly, IDF has really similar performances for some event categories while performing really poorly in some others. In terms of Sports or Arts, Culture & Entertainment, the percentage of detected events is approximately the same as the other models. However, for other categories such as Armed Conflicts & Attacks, IDF performs much worst than the other models.

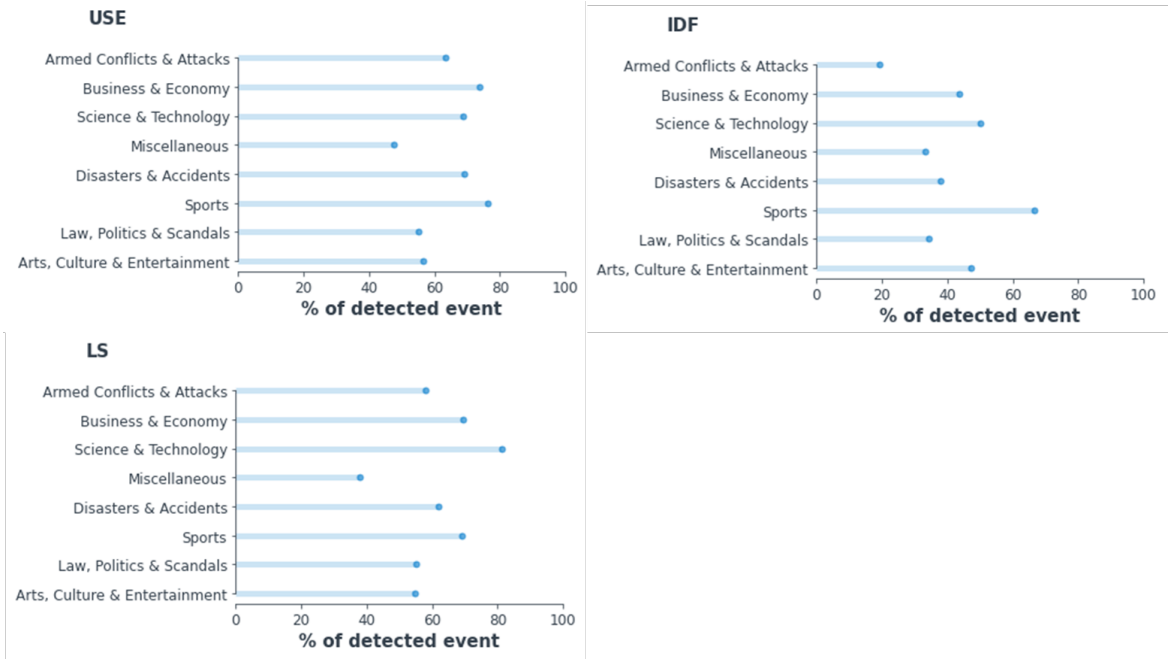

Figure 5.9: Percentage of detected events according to the category for each model.

The last aspect we want to analyze is the time taken to detect the events. It is illustrated both by TPR and Figure 5.10. To obtain this figure, we compute the time taken to detect an event, i.e. the difference in terms of hours between the window in which the first labeled tweet corresponding to the event appears and the window in which the event is detected and annotated as "True" by our evaluation method. As we can see, the general trend seems to be the same for all three models. USE and LS have a steeper curve but it corresponds to the highest number of events they detect. Overall, the models seem to detect with the same pace, according to their performances.

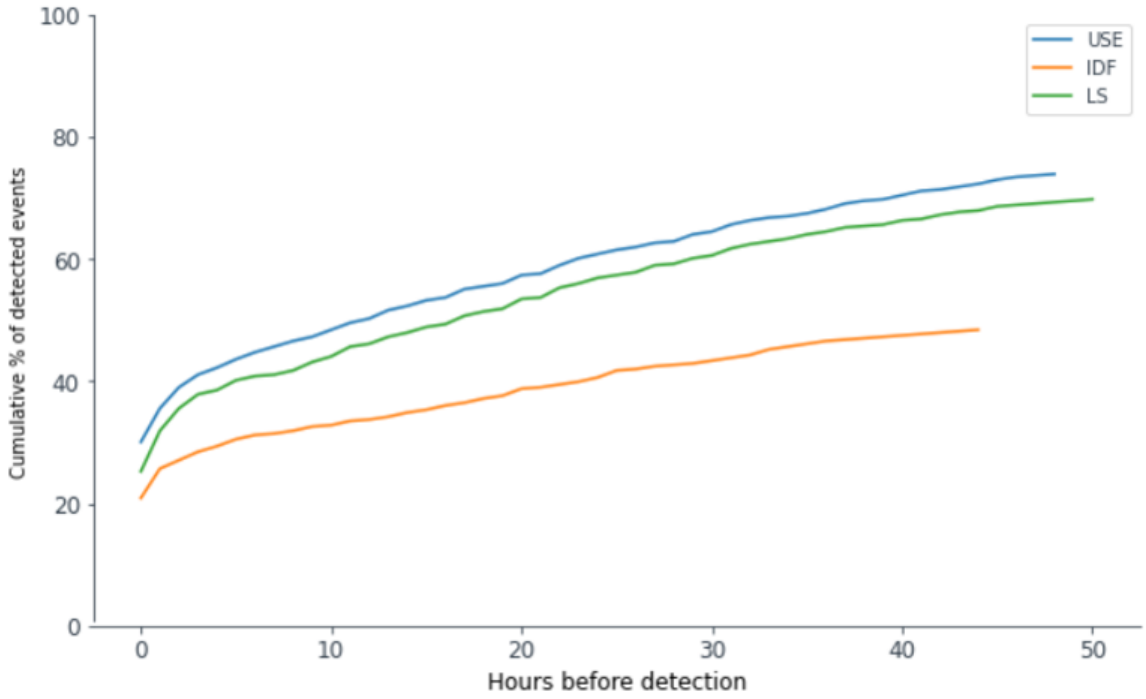

Figure 5.10: Representation of the time taken by each model before detecting the events. Using the model USE, about 50% of the events are detected less than 10 hours after the publication of the first corresponding labeled tweet.

Partial Conclusion

In this part, we evaluated our event detection system on the whole dataset using different text representation models. Overall, the same conclusion can be drawn compared to the evaluation conducted on the annotated part. An interesting experiment would be to evaluate different methods for the "Event Detection" module, to determine whether this affirmation can be generalized or not. However, we decided to leave this part for future work and focus on other modules of the system instead of conducting such experiments.

5.4.2 Event Tracking

In this subsection, we first evaluate the performances of different representations for the clusters on the annotated dataset to chain them and track them over time. Then, we apply the best method to the full dataset.

Evaluation methods

We evaluate different aspects of the chains. The first that is evaluated is whether all the events of the chain are associated with the same ground truth event. If one of the detected events is associated with a different ground truth event than the other detected event of the chain, then the chain is considered false.

Table 5.8: Results from chaining using the entity-based method. We used a threshold of t=0.55. In the ground truth, there are 373 chains.

Method	Count true	Count false	% of chains found	% of coverage
Entity based	183	45	0.49	0.26

For the chains that are labeled as true, we evaluate the coverage of the event. For each ground truth event, we determine in which window it starts, in which window it ends, and we determine whether the event is continuous or not. Then, we compute in which window the corresponding detected event is detected and compute the coverage.

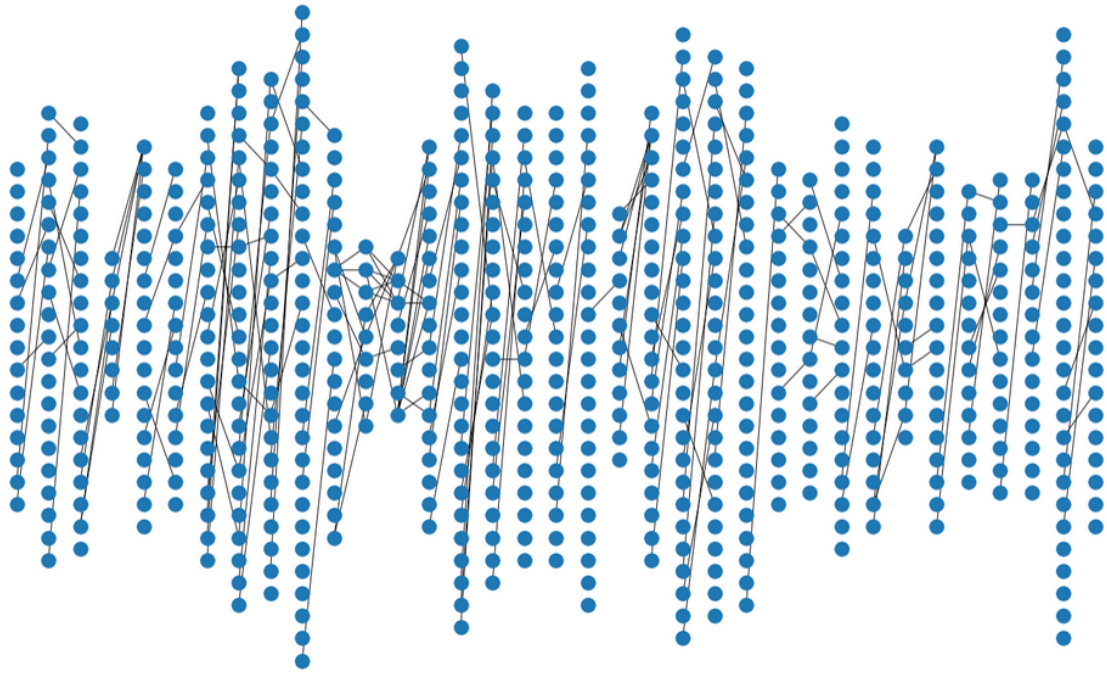

Figure 5.11: An example of a cluster chain obtained using our methods. Each column is a window, each dot is an event. For the sake of visualisation, we use windows of $\tau = 2000$. We represented the clusters using entity-based representation.

Results

We evaluate each of the representation methods using different thresholds value to determine which one performs better. As we can see in Figure 5.13, both of the values based on documents perform very poorly. Their coverage is nonexistent and they create multiple false chains. The method based on entities is performing correctly. The number of wrong chains can be really low for high threshold values however it also reduces the number of correct chains and coverage. Thus, a threshold value around 0.5 seems to be the better choice. We give further details about this method in Table 5.8.

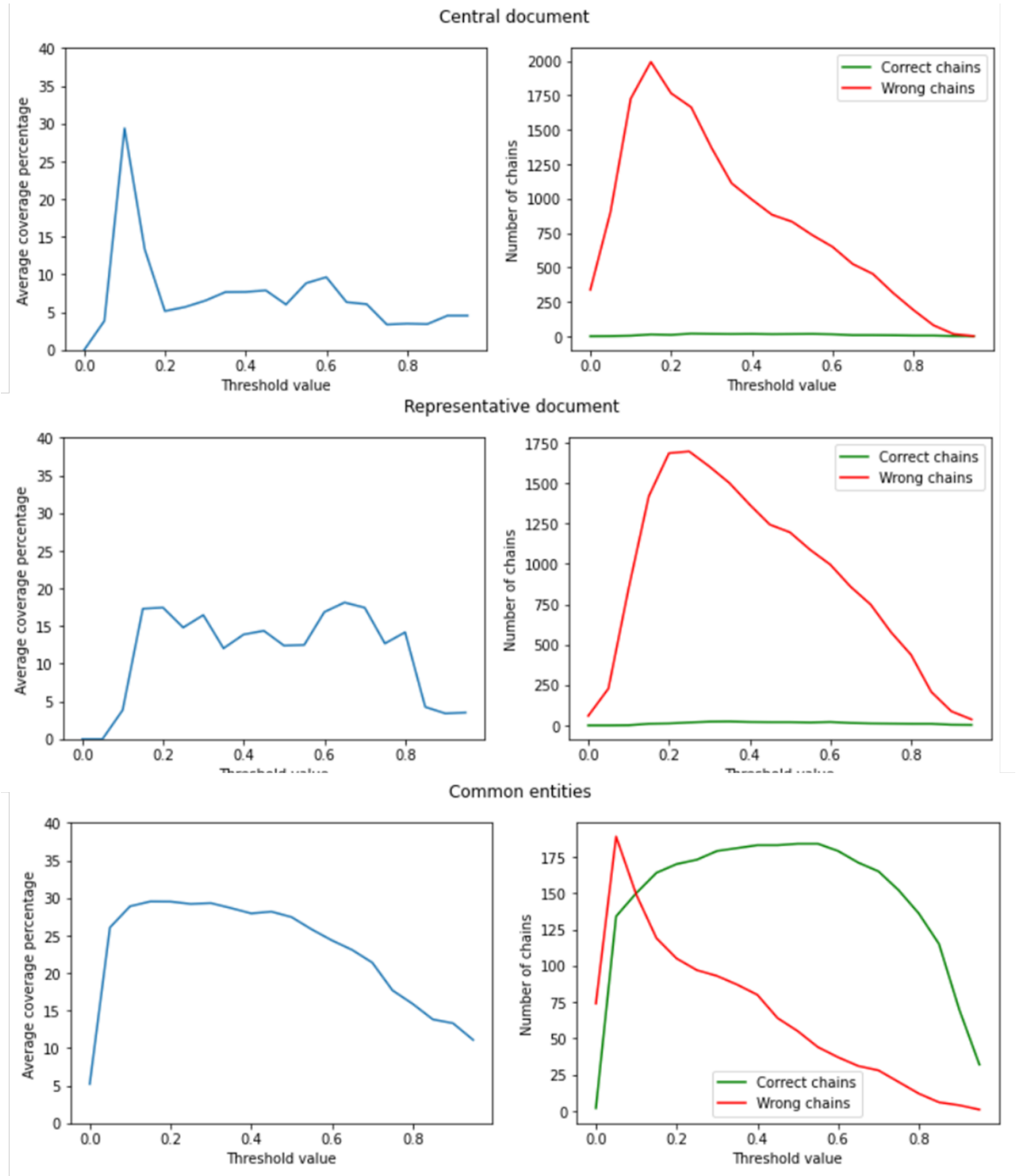

Figure 5.12: Evaluation of the performances of each representation method for varying thresholds. As we can see, the methods based on representative or central documents do not perform at all (please note the difference of scales between the right-hand side figures). The method based on entities has decent performances.

Table 5.9: Results from chaining using the entity-based method on the full dataset. We used a threshold of t=0.55.

Method	Non predicted	Non fully predicted	False	Correct
Entity based	3489	634	20	105

Analysis

The only method which performs correctly is the method based on the entities. The major issue related to the method based on a representative or central documents is that it creates too many chains and most of them are wrong. We think it is because selecting only one tweet is not discriminative enough and the content embedded in this representation is not enough to depict multiple aspects of the clusters. Indeed, the method based on entities is the only one that considers a result linked with the overall content of the cluster and not only one of its elements. Thus, it is why the performances are the best.

Application to the whole dataset

We apply the same method but this time to the whole dataset. We divide the results for each cluster chain into four categories:

- **Non predicted**: No cluster is in ADE, i.e. there is no cluster in the chain that contains at least 1 annotated tweet. As explained earlier in the evaluation method part, we cannot evaluate these chains other than with human annotation.
- **Non fully predicted**: there is at least one clustering ADE in the chain, but there are also some clusters without any annotated tweet.
- **False**: all the clusters of the chain are in ADE, but at least one of them is annotated as false, i.e. there is no match between this cluster and the ground truth event, as we defined in the evaluation section.
- **Correct**: all the clusters of the chain are in ADE and in MDE.

We present the results in table 5.9. As we can see, the results are promising, with a lot of correct chains and only a few false. However, there are a lot of NP and NFP chains. We focus on the NFP category and try to analyze the results in order to have more insights about the performances of the system.

Some results are displayed in figure 5.13. Most of the chains are composed of a few clusters in ADE as we can see from figure 5.13 (a). In figure 5.13 (b), we want to evaluate whether chains composed of a lot of events from ADE have more elements from MDE than the other chains. Even if this seems to be true for most of the

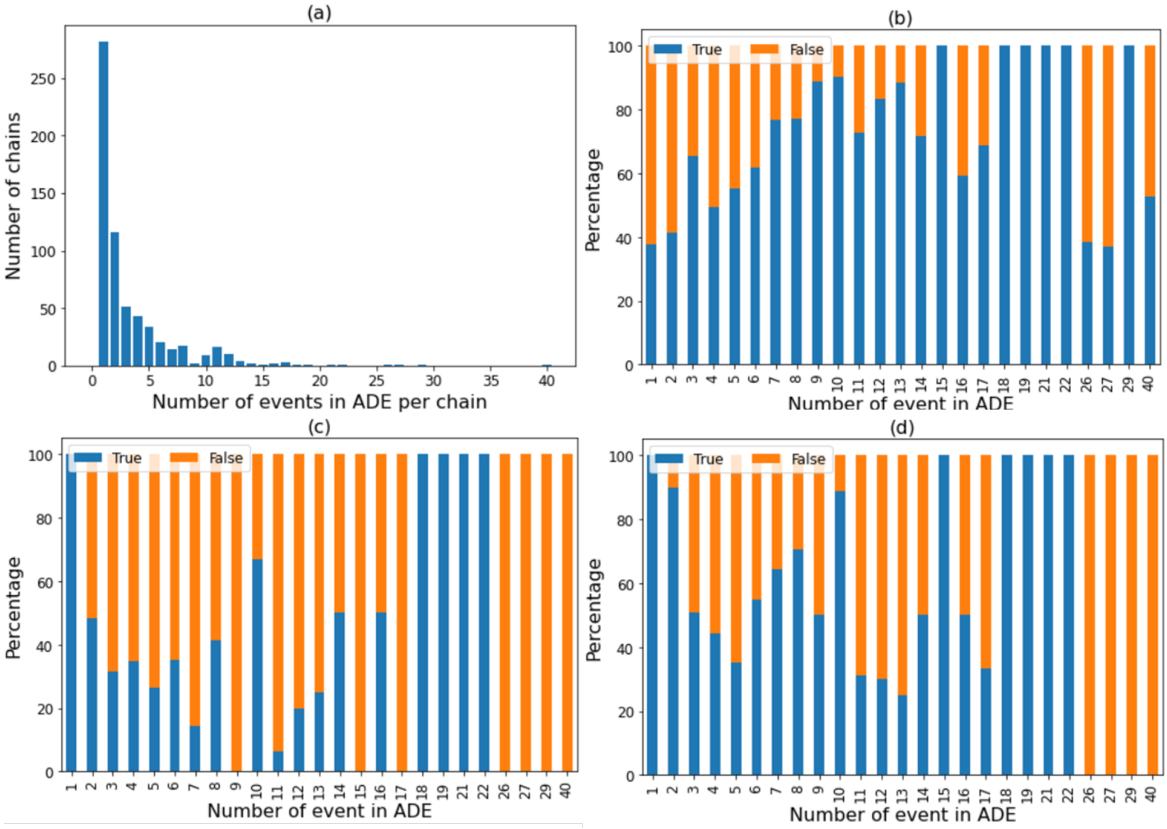

Figure 5.13: Evaluation of the quality of the obtained event chains on the whole dataset. (a) Illustrates the number of events of ADE in each chain. (b) Illustrates the percentage of events that are in MDE in each chain. In figure (c), we evaluate whether the elements of ADE in each chain are associated with the same event. Finally, figure (d) illustrates whether the elements of ADE are associated with events from the same category.

chains of 8 events of ADE, this is contradicted by the chains containing 26 and 27 elements from ADE. Overall, it seems reasonable to think that chains containing a lot of elements from ADE contain more elements from MDE than other chains. In figure 5.13 (c) and (d) we want to evaluate whether the chains effectively deal with the same event or not. Figure (c) illustrates the percentage of chains that contains only elements from ADE associated with the same event. As we can see, it is mostly not the case, and different events are associated together. In Figure (d), we evaluate whether the events in the same chain are from the same category. The obtained results are overall better, meaning that even if the chains do not only contain the same event, they tend to deal with similar subjects, indicating possibly interesting performances.

5.4.3 Partial conclusion

The experiments of section 5.4.2 showed that only the method based on entities' similarities is efficient to measure the similarity between the event clusters. We think it is coherent with definition 2.1.1, particularly because "[An event] is identified by a group of entities (e.g. people; location) that is discussed in the documents dealing with the event".

In the last part of this section, we evaluated the performances of the chaining method on the whole dataset and faced the same difficulties, considering the lack of labeled documents in the dataset. We provided some insights about the performances with an evaluation similar to the evaluation method introduced in the previous section but it is difficult to draw meaningful conclusions about the actual performances.

5.5 Comparison of EDS to other event detection systems

To complete the analysis of our event detection system, we compared its performance to the performances of other detection systems in the literature. We considered comparing EDS [4] to 3 other systems : MABED (Guille and Favre, 2014), an adaptation of the FSD algorithm (Mazoyer et al., 2020a) and Embed2Detect (Hettiarachchi et al., 2021). However, the experiments showed that Embed2Detect is not adapted to the task we want to perform. Indeed, the output of this system is event windows, each described by a set of keywords. These keywords are not grouped, meaning that it is not possible to understand which keyword corresponds to which events. Thus, the results are not comprehensible to a human being in our context. A sample of the output obtained on the Event2012 dataset with this system is provided in table 5.10. The other systems are presented in section 2.4, but we give a summary hereafter. We chose these systems according to different criteria: We tried to compare our system to both feature-pivot and document-pivot systems, to have good representativeness of

[4] https://gitlab.com/Emaitre/eventdetectionsystem

Table 5.10: Sample of the result obtained for an event window of the Event2012 dataset using the Embed2Detect system. This system is not adapted to the detection of multiple overlapping events. Some words seem event-related and some are not informative. As none of the words are grouped, it is difficult to understand which events are happening.

8am	drive	james	speakers	test	wedding	annual	premiere
pet	kick	shine	stay	classic	green	respect	justin
killing	breakfast	break	gym	later	mornings	fight	wednesday

what exists in the literature. Unfortunately, Embed2Detect was the only feature-pivot approach we used so no such approach appears in the comparison. We used systems that are available online and tried to use them according to the recommendations of the authors.

5.5.1 Description of the systems

FSD

We use the same algorithm as in chapter 4, however, it has no event detection step meaning that all the clusters are considered as events. Thus, we apply the same event detection step as in EDS, presented in section 5.2.2. Just like EDS, it is a document-pivot approach. It group the tweets by clusters. Thus, the content of the cluster can be analyzed for event detection and the evaluation of the system. We used the python implementation proposed by the authors [5] and completed it with the event detection step.

MABED

MABED (Guille and Favre, 2014) is a document-pivot approach that relies solely on tweets and leverages the creation frequency of dynamic links (i.e. user mentions) that users insert in tweets to detect significant events and estimate the magnitude of their impact over the crowd. MABED dynamically estimates the period of time during which each event is discussed. It provides a textual description of the events, namely the main word and a set of weighted related words. It then ranks the events according to their impact. Thus, the output of the algorithm is not a cluster of documents but some words describing each event. The system does not keep track of the origin of the words, meaning that we do not know which tweets are associated with each event. We used the python implementation proposed by the authors [6].

[5] https://github.com/ina-foss/twembeddings
[6] https://github.com/AdrienGuille/pyMABED

5.5.2 Experimental configuration

Hardware

We all the experiments on a machine running on Ubuntu 18.04, with a Bi Xeon Silver 4208 processor, 192GB of 2 933Mhz RAM and a NVIDIA 1080TI running CUDA v11.

Dataset

We used the same dataset as for the previous experiments, namely Events2012 (McMinn et al., 2013). For all three models, we considered a set of 57 million tweets. For EDS and FSD, applied the filtering rules and cleaning steps described in **??**, to obtain a set of 16 million tweets.

For MABED, we conducted two experiments. For the first experiments, we applied no filtering steps and adapted the cleaning steps to keep the user mentions in the tweets because MABED uses them. In the second configuration, referred to as MABED_filtered in the following section, we filtered the tweets in the same manner as for FSD and EDS but adapted the cleaning step to keep the user mentions.

Systems parameters

For each system, we chose the parameters recommended by the authors:

- **FSD**: we chose TF-IDF with an IDF calculated on the whole dataset as the text representation model. We chose this model because it is the one that performs the best using the FSD algorithm. We used $t = 0.75$ as threshold value, as recommended. The FSD does not need a parameter about time window duration.

- **MABED**: the system requires different parameters. We chose a time window of 1 hour, just like the configuration of EDS. For the other parameters, namely p the number of words per event, θ the minimum magnitude of words describing an event, σ the threshold value for event merging, maf the minimum absolute word frequency and mrf the maximum absolute word frequency, we chose the default values provided by the authors : $p = 10$, $\theta = 0.60$, $\sigma = 0.60$, $maf = 0.4$, $mrf = 10$.

Evaluation

To evaluate the event detection systems, we apply the evaluation method described in section 5.3. The method is directly applicable for the FSD, so we follow the exact same steps described in this section.

However, for MABED, the method cannot be directly applied. Indeed, the events are only constituted of keywords and the system does not associate the tweets to an

Table 5.11: Results of the manual annotation for the FSD algorithm. We manually annotated 100 events., We also show the results of the annotation for EDS for an easier comparison

System	Precision	Recall	F score	DER(%)
FSD	0.1	0.02	0.04	0
EDS	0.85	0.42	0.56	2

event or keep track of them. Thus, manual annotation is necessary. MABED ranks the events by impact, so we manually annotated the 200 more impacting events to estimate the performances.

5.5.3 Results

Generalities

MABED detected a total of 732 304 events. MABED filtered detected a total of 405 136. On the other hand, FSD detected 9 509 events.

In terms of time performance, MABED completed the task in 64 335 seconds, meaning the system treat approximately 886 tweets per second. MABED_filtered performed the task in 111 949 seconds, due to the time of filtering, which means approximately 510 tweets per second. The running time of FSD was approximately 9 days, including the filtering and the event detection tasks, which corresponds to 73 tweets per second. Finally, the full EDS running time (including filtering) is 159 459, which means approximately 357 tweets per second.

On Twitter, approximately 6 000 tweets are posted every second. However, only 1% of these tweets are accessible using Twitter's API. Thus, an event detection system should be able to treat 60 tweets per second to treat the stream in real-time. Both these systems satisfy this condition.

Human annotation

For the FSD algorithm, the results obtained are summarized in table 5.11. We manually annotated 100 events following the method presented in section 5.3.

Concerning the results of MABED, they are presented in table 5.12. We estimated the precision by annotating the 200 most impacting events and labeling them as event-related or non event-related. Out of the 200 events, 112 of them were event-related. In terms of recall, estimating the value is more difficult. We annotated 59 events as related to either the 2012 US election or hurricane Sandy. Including these two events and the other duplicated events, we identified 41 different events. Out of these 41 events, we mapped 25 of them with events from the ground truth of the dataset. The

Table 5.12: Results of the manual annotation for MABED. We manually annotated the 200 more impacting events.

Model	Precision	# unique events	# matches with GT	DER(%)
MABED	0.56	41	25	24
MABED filtered	0.60	51	29	27

Table 5.13: Results of the evaluation method for the FSD algorithm. We manually annotated 100 events. We also show the results of the evaluation applied to EDS for an easier comparison

System	# of events	Precision	Recall	F score	DER(%)
FSD	26	0.08	0.05	0.07	27
EDS	323	0.68	0.65	0.66	8

easiest measure to compare between EDS and MABED is precision. As we can see, EDS achieves a precision of 0.85 while MABED achieves a precision of 0.60 at best. Other measures are difficult to compare because for EDS, we annotated the events in a time-ordered manner, meaning that it has seem less events than MABED, thus it has most likely detected less events than MABED.

Application of the evaluation method

For the FSD algorithm, we obtain after training an optimal threshold $t = 0.05$. We show the results in table 5.13

Discussion

As we can see from table 5.11, EDS performs better than FSD in this context. The FSD algorithm seems to be really sensitive to the noise we introduced by using the whole dataset. During the annotation performed during the evaluation procedure, we noted that the only events correctly identified are events containing only a few tweets (most of the time < 100). On the other hand, events with a lot of tweets (sometimes with more than 100000 tweets) usually contain a lot of noise and uninteresting events. These events are most of the time associated with ground truth events such as the presidential debate during the 2012 US elections. We believe that these detected events, spanning multiple days due to the continuous stream of new tweets, contain tweets that are too diverse and thus are likely to be in the nearest neighborhood to a new tweet. Thus, they keep growing and become super-clusters containing both interesting tweets and noise, making them impossible to understand. We think it is a strength of our approach which clearly discretizes the stream of documents and thus

limit the risk of creating super-clusters vacuuming too many documents. To mitigate this risk, rising the threshold value of the algorithm may be desirable. Considering the prohibitive running time of this system, we did not perform another run to test this hypothesis.

Concerning MABED, table 5.12 summarizes the results. We could only evaluate the precision, the number of detected events, and the number of detected events related to the ground truth because the system does not keep track of the original document of the features and does not associate any document with an event. Thus, we could not apply our evaluation method. The results obtained by MABED are quite good and seem to be a good compromise considering the efficiency in terms of the running time of the method. However, we think that the precision might decrease by annotating more documents because MABED ranks the events by impact and thus the best events should be displayed first.

Overall, EDS achieves good performances while keeping a decent running time. Thus, we think that EDS is competitive with these event detection systems.

5.6 Conclusion

In this section, we first presented the modules related to phase 3, namely event detection and event tracking and summarization to obtain the full version of EDS. We evaluated the performances of the event detection phase using a new evaluation method for event detection systems, applicable to existing datasets. This method greatly reduces the need for human annotation and allows reproducibility of the results. We then evaluated different representations of the clusters for the event tracking method and evaluated the best method on the full version of Events2012. Finally, we compared the performance of EDS to other event detection systems in the literature and showed its competitivity.

Overall, evaluating event detection systems in a real-world setup is challenging. The new evaluation method we proposed allowed us to draw some interesting insights into the system. However, we also pointed out that the method is not necessarily applicable to all the event detection models as it, notably to systems that do not keep track of the original documents or do not associate any documents with the detected events. Despite this, we think that this method is an interesting step toward reproducibility of the results.

In the next chapter of this work, we present an application of this event detection system linked with the industrial context of this thesis.

Chapter 6

Application to the context of raw materials

In the previous chapters, we have extensively developed our work on event detection, which is the core of this thesis. EDS, our event detection system, is intended to be applied to an industrial context, linked among other things to the supply chain and stock markets of raw materials. In this chapter, we develop the different aspects of this application and why Scalian is interested in this type of application.

In particular, we investigate how EDS can be integrated into a framework involving synergy with the business which will help decision-makers by providing insightful information about events impacting raw materials supply chains and prices in real-time. This synergy is especially important when it comes to identifying the information to look for, and the sources to extract this information. In the literature, there is a lack of an investigation about the data sources to monitor to detect the impacting events that might disrupt supply chains as well as about the nature of the events to supervise. In this chapter, we first propose an integration of EDS into such a framework. Then, we identify the relationship between the historical events and the variations of the commodities stock market to calibrate the event detection system filters, i.e. establish which type of event to monitor (political, weather conditions, geographical localization ...) and which sources to focus on. Thus, this chapter focuses on the issues related to phase 1 of the event detection framework.

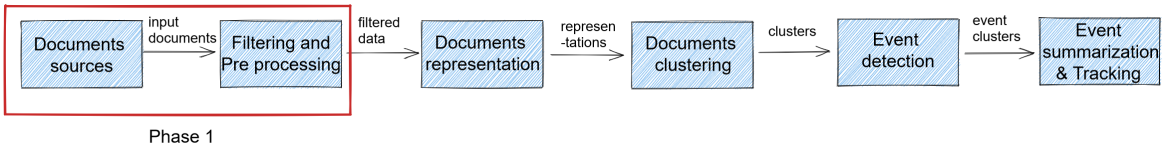

Figure 6.1: Phase 1 of the event detection framework

6.1 Introduction

The commodity market is the basis of many industrial and consumer goods supply chains (SC) around the world, being responsible for moving 30% of the world's goods (Radetzki and Wårell, 2016). The market volatility has been particularly addressed by different research efforts because of its intrinsic complexity of products, producers, geopolitical, economics and weather conditions, etc. (Huchet-Bourdon, 2011). Considering the tremendous amount of information exchanged online, it is not easily possible to keep track of everything. However, these external events can have a direct impact on supply chains and raw materials prices but are currently underexploited since existing IT solutions are not able to address this problem (Leveling et al., 2014). The authors of (Fan et al., 2015) provided a novel direction on using Big Data and Machine Learning technologies in supply chain management and highlighted the complexity and need of using advanced analytics for external uncertainty analyses. This way, the number of unpredictable low-frequency and high-impact events, also called "black-swans" by (Nicholas Taleb, 2015) can be reduced with these techniques.

We have developed an event detection system that has been presented in the previous chapters. To detect the different events that can potentially be impacting, it is necessary to detect general events as well as weaker events that are more specific to each domain. Thus, developing filtering methods are necessary to focus on these specific domains and capture information that would be drowned in the stream otherwise. In the literature, there is a lack of an investigation about the data sources to monitor to detect impacting events and about the nature of the events to supervise. Thus, we are interested in conducting a study that answers these questions. To do so, we propose to conduct a historical study of events that impacted the raw materials stock market in the past. Our idea is that the kind of events that were impacting in the past can be interesting insights to identify which events to supervise. However, this is not sufficient to have a representative vision of which events to look for. Indeed, some events have no equivalent in the past, such as the Suez canal congestion or the covid-19 pandemic. Thus, it is necessary to consider the expertise and the feedback of the end-users, namely raw materials buyers and supply chains managers, and to involve them in the process, to ensure that we do not miss any important events.

In the first section of this chapter, we present the method used to conduct this work and the integration of EDS in such a framework. In a second section, we present some details about the context and a related work section, in which we introduce some elements about raw materials to better understand which types of raw materials exist and their characteristics. We also present some of the factors influencing the commodity stock market and we justify why external sources such as Twitter can be interesting for the anticipation of raw material prices variations. Then, we present why we decided to focus on a particular commodity, the phosphate, and some of the

factors impacting this commodity. After this related work and context section, we conduct a historical study of the raw materials stock market and we analyze which are the impacting events, causing raw material stock variations. We apply these results to a case study about phosphate. Finally, we propose a primary experiment on Twitter's data stream. The work presented in this section was published in (Maitre et al., 2022) and presented as a poster at ECIR 2022 Industry Day[1]. We would like to thank Giovana Ramalho Sena for her precious contribution to this section.

6.2 Method

To validate the interest of this multidisciplinary approach (computer science, machine learning, supply chain management, economics) and its applicability in the fields of logistics and procurement, we conducted at the beginning of this thesis a series of interviews with experienced supply-chain experts and purchasers in addition to the review of the literature. In total, we consulted five different experts throughout 10 hours of in-person/phone interviews. The results of these interviews served as a guide for this research, by providing the firsts insights for the following questions:

- What type of data sources should we use?
- What type of signals should we monitor?
- How to define a signal?
- Which metrics should we use?

The output of these interviews revealed that information sources are multiple and they are difficult to follow. All the experts agreed that events have an impact on raw materials' price, hence the choice of focusing on event detection. However, an in-depth study of the dynamics of variation is needed for each commodity. Considering the wideness of the domain and the multitude of raw materials, we concluded that a pilot study had to be performed to assess the feasibility of this approach. The Phosphate was then chosen to perform the rest of the study since it has been included as a Critical Raw Material by the European Commission, the absence of substitutes, the fact of being a mineral needed for the food system, and of particular scientific interest. More details about this choice are presented in section 6.3.4.

The framework we propose is divided into two major components, the (A) business component and the (B) IT component, i.e. EDS our event detection system, which is interdependent and complementary. The general architecture is described in Figure 6.2. The former, which will also be assimilated as phase 0, is an in-depth study of the events that historically impacted the raw material to be supervised, in our case

[1] `https://ecir2022.org/industry-day/`

the phosphate. These results will then be exploited to calibrate the filtering rules of the first phase of our event detection system, which was extensively presented in the previous chapters of this document.

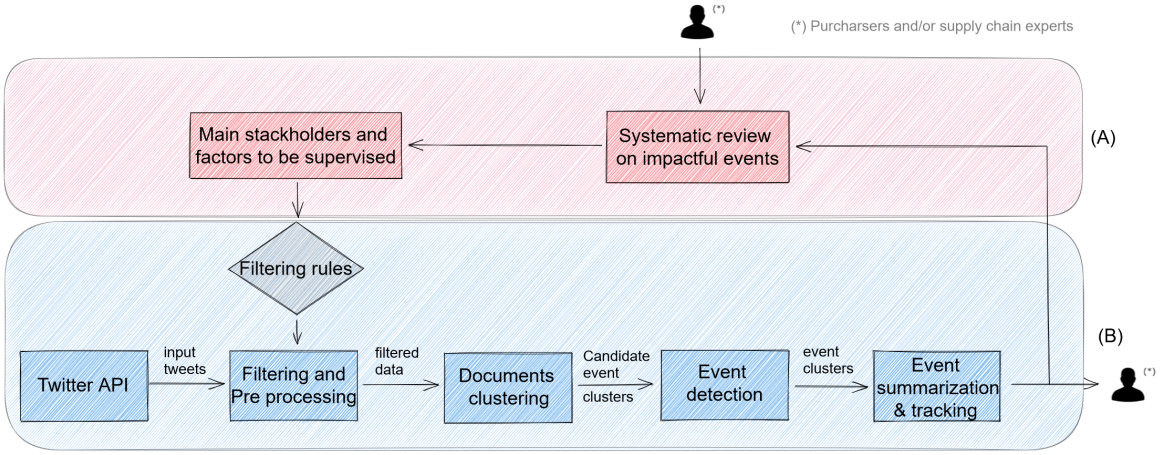

Figure 6.2: The architecture of the proposed framework. The red part, annotated (A), corresponds to the business component and the blue part, annotated (B), to the IT component, i.e. EDS, our event detection system.

6.2.1 Proposition for impacting events mapping

To create the data filtering process in Phase 1, it is necessary to identify the events that had historically impacted raw materials prices. Using these results, we can focus on potentially interesting domains, avoid processing too many documents and only consider the relevant ones by using the appropriate filters. As this phase is a building block for EDS robustness, it is so-called Phase 0.

Hence, a literature review was conducted to create a typology of events that have already been identified as triggers and/or reactions to a price change. The following procedure was then executed for the phosphates case study and could be replicated for other raw materials. In addition, the main actors and their geographical zone were identified to further monitoring, by the analyses of resources and production distribution with data collected from the U.S. Geological Survey, Minerals Yearbook 2018, v.I, Metals and Minerals, as well as an import and export analyses with World Bank data.

To respond to the multidisciplinary needs of this research, Google Scholar was the main search engine used since it indexes a multitude of sources. Most consulted publications were in peer-reviewed journals in the fields of Raw Materials, Resources, Economics, and Food Policy. In addition, conference proceedings and books, as well as dissertations, scholars working papers, and institutions' publications (i.e.: OCDE, World Bank) were selected to enrich the analyses. The following keywords were used: commodity, raw materials, price, supply chain, events, and phosphates. Due to the

recent emergence of social media which modified the classical ways of communications, papers published after 2000 were privileged.

Therefore, a non-exhaustive list of the main causes of perturbations in raw materials price was identified and they are presented in a summary table in section 6.4, together with their literature of reference. These causes can be used to setup the filtering rules to detect events that are similar to those which have been identified in the literature. Then, a list of the main actors of the supply chain was elaborated, associated with the role, geographical zone, and impact characteristics. These lists have to be consistently updated, using the feedback given by the event detection model and the analysis of experts. The quantification of the events in terms of price variation intensity, most probability, and their importance is not treated in this chapter, being subject to further investigation.

In the next sections of this chapter, we present the application of this method, starting with a contextualization.

6.3 Context

The objective of this section is to give more details about the raw materials, the factors that influence their prices, and, also why we decided to use social media as our principal source of data. We also provide present why we chose to focus on phosphate and what factors influence its price.

6.3.1 Definition

The Oxford Dictionary defines raw materials as "The basic material from which a product is made". On Wikipedia[2], we can find a more complete definition "A raw material, also known as a feedstock, unprocessed material, or primary commodity, is a basic material that is used to produce goods, finished products, energy, or intermediate materials that are feedstock for future finished products.".

Different types of classification exist in the literature. The German Federal Institute for Geosciences and Natural Resources has proposed the first classification. This classification is based on the usage of the raw material and is presented in Figure 6.3.

In this classification, raw materials are divided into two major categories: soft commodities and hard commodities. Soft commodities come from living nature (animals and plants), very often used and produced by the food industry, being able to regenerate in the short term. Hard commodities, on the other hand, do not regenerate on a human scale, leading to a risk of depletion of these types of resources. These

[2]https://en.wikipedia.org/wiki/Raw_material

[3]`https://www.postfinance.ch/en/private/needs/investing-in-simple-terms/how-can-i-invest-in-commodities-.html`

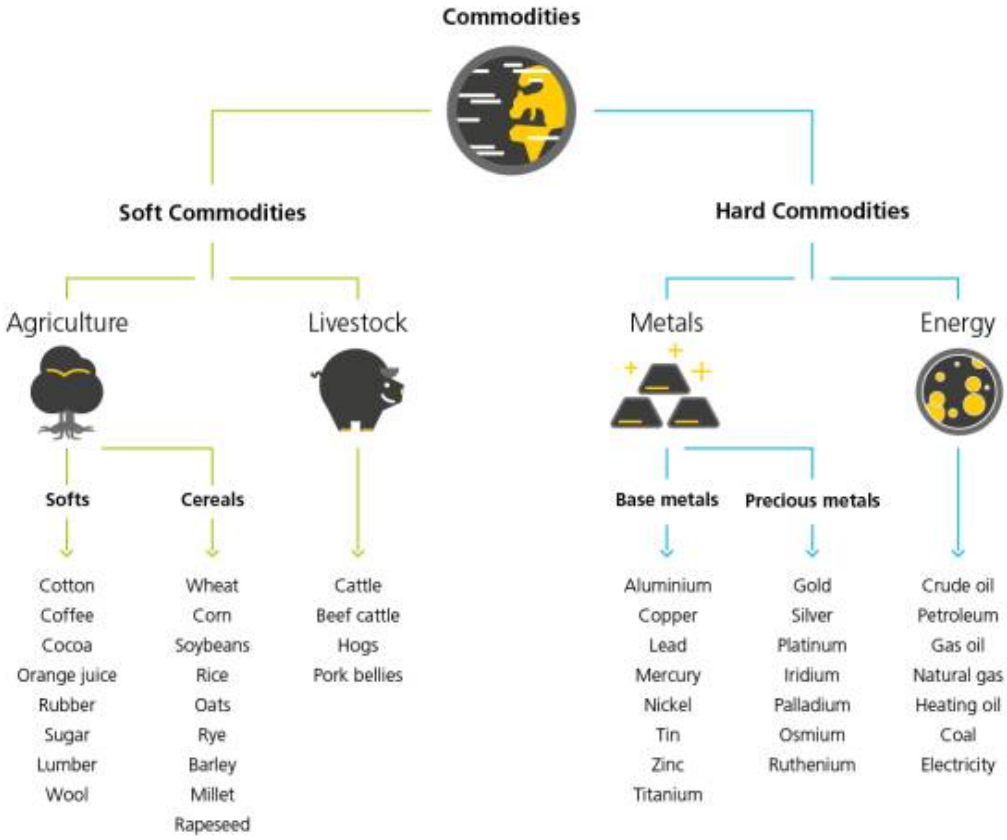

Figure 6.3: Classification of the commodities. Source[3]

resources include metals, minerals, or energy-related commodities. This classification allowed the understanding of the renewable or non-renewable characteristics of the commodities.

Thus, for a matter of definition, raw materials are often considered the output of the primary sector, comprising agriculture (hunting, forestry, and fishing), mining (including fossil fuels), and utilities. However, according to (Radetzki and Wårell, 2016) this definition is narrower and a broader and more used classification uses the Standard International Trade Classification (SITC) [4], designed by the United Nations, sections and divisions, allowing to find intrinsic characteristics of groups. An example of a section is presented in Figure 6.4 This division takes into consideration group (A) for "Food in a broad sense" referred to SITC Section 0 + 1 + 22 + 4, group (B) for "Agriculture commodities" Sections 2 – 22 – 27 – 28, group (C) for "Minerals and Metals" Sections 27 + 28 + 67 + 68 and finally, group (D) for "Mineral fuels" referred to section 3.

Section 2 – Crude materials, inedible, except fuels		**36**	**115**	**239**
Hides, skins and furskins, raw	21	2	7	11
Oil-seeds and oleaginous fruits	22	2	10	12
Crude rubber (including synthetic and reclaimed)	23	2	5	16
Cork and wood	24	5	13	18
Pulp and waste paper	25	1	7	14
Textile fibres (other than wool tops and other combed wool) and their wastes (not manufactured into yarn or fabric)	26	8	23	48
Crude fertilizers, other than those of division 56, and crude minerals (excluding coal, petroleum and precious stones)	27	5	17	45
Metalliferous ores and metal scrap	28	9	24	43
Crude animal and vegetable materials, n.e.s.	29	2	9	32

Figure 6.4: An example of a section from SITC.

6.3.2 Factors influencing the commodity market

According to (Radetzki and Wårell, 2016), in the short term, the balance between supply and demand defines the price of commodities. Classical microeconomics assumes that disturbs in the supply are caused by a change in the production costs, mostly due to technological advances, taxes, and subventions. The consumer's income, price of substitutes and/or complementary products, and the number of buyers influence demand (Pindyck et al., 2013).

Despite the presence of different materials in the groups, most of the substitutes for each material belong to the same group, which is more evident in the fuels group. (Radetzki and Wårell, 2016) premise that for agricultural products, groups (A) and

[4] https://unstats.un.org/unsd/publication/SeriesM/SeriesM_34rev4E.pdf

Group	Group Name	SITC Section
A	Food	0 + 1 + 22 + 4
B	Agriculture commodities	2 - 22 - 27 - 28
C	Minerals & Metals	27 + 28 + 67 + 68
D	Mineral fuels	3

Table 6.1: Commodities classification according to (Radetzki and Wårell, 2016)

(B) of the table 6.1, price instability is often caused by disturbances on the supply side and the cultivation of certain products is highly geographically concentrated. However, price fluctuations for minerals, (C) and (D), are usually on the supply side - notwithstanding strikes and cartels.

Concerning the question of the increasing price volatility among commodities during time, (Jacks et al., 2011) presented empirical evidence from 1700 to the present indicating that commodity prices have historically been more volatile than those of manufactured goods have. The same study finds that volatility has not increased over time; on the contrary, globalization and the integration of the world market have led to less volatility compared to situations of economic isolation.

Another raising question is the relationship between mineral fuels price peaks and resource depletion. (Henckens et al., 2016) investigated the relationship between the price trend of mineral raw materials and the availability of its resources for future generations, by analyzing the market mechanisms. He concluded that despite the fluctuation in mineral resource prices, there is no significant correlation with resource depletion, but with the balance dynamics of supply and demand.

Finally, (Anani, 2019) studied the long-term sustainability of countries relying exclusively on the commodities market. The author studied the price dynamics from the angle of the limits of the Hotelling rule (Hotelling, 1931) and defends that the supply-demand balance dictates commodities price, together with the technical progress and business structure. Plus, the study highlights the growing influence of speculation due to the financialization of the raw materials market.

The main findings are that although commodity prices are volatiles, a balance between supply and demand usually settles them. Supply and demand are affected by internal and external events that can disturb the system balance and then influence the price. For the first, disturbs are mainly due to a variation in production costs and for the latter, the price of substitutes and/or complementary products, the number of buyers, and their income. In addition, some macroeconomic conditions, such as speculation and energy prices also affect the price.

In the literature, there is a lack of an investigation about the data sources to monitor to detect the impacting events and about the nature of the events to supervise.

In the next section, we study how the variations of the stock markets can be linked to activity on social media.

6.3.3 Social media and the stock market

The choice to monitor social media to anticipate variations in the stock market is not necessarily obvious. In this subsection, we highlight that different works from the literature showed promising results, justifying our interest in this approach.

In (Bollen et al., 2011), the authors use fuzzy neural networks to forecast Dow Jones based on the mood of people on Twitter. They use 2 tools to understand moods: OpinionFinder and GPOMS. They compare it to the daily time series of DJIA (Dow Jones Industrial Average) closing values. They use a Granger causality analysis in which they correlate DJIA values to GPOMs and OF. The results are satisfying to the authors. The authors of (Zhang et al., 2011) try to predict stock markets such as Dow Jones, NASDAQ, and S&P 500 by analyzing tweets. They measure fear and hope every day by measuring the frequency of words carrying emotions such as "fear", "worry", and "hope". They count the number of tweets containing these words and work with the percentage of daily usage of these words, comparing it with the market indicators of the next day. Emotional tweets are negatively linked with the stocks. Every emotion is correlated with a drop in the stocks. It means that emotion is correlated with incertitude, no matter if the emotion is positive or negative.

In (Chen et al., 2014), the authors pursue an analysis of the discussions around articles posted on social media such as Seeking Alpha[5] which are specialized in finance. They study advices given online by investors to investors (not necessarily professionals). The hypothesis is that advices given online contains interesting information. They study the tone of the documents and say that the literature says that negative words capture the tone of the document. They compare the tone of the documents with the tone of the commentaries. If there is a disagreement, usually the commentaries are right. Finally, they study the relationship between the history of an author and the commentaries. If the author has a bad reputation, the commentaries are more often negative and they tend to be right most of the time.

The authors of (Oliveira et al., 2016) argue that the analysis of social media data may allow a deep understanding of users' behavior: their sentiments, identification of their interests, measurement of users' influence. There is a strand of the finance literature (behavioral finance) that argues that sentiment may affect financial prices (Shiller, 2003).

The authors of (Burnie and Yilmaz, 2019) propose an analysis of the change in discussions on social media with bitcoin price: the authors temporalize word2vec to detect the most discussed topics during certain phases of the bitcoin time series. They conclude that certain types of vocabulary can be associated with a defined area of the price time series.

Thus, several studies attempted to link stock market variations with social media

[5]https://seekingalpha.com/

activity. We believe that predicting the stock market is an excessively ambitious task. However, assisting experts in their decision-making by providing them insights seems to be a more manageable task. This is what we intend to do by combining their expertise and our event detection system in the framework presented in this chapter.

In the next section, we present why we chose phosphate and the different steps of the project that led to this decision.

6.3.4 The phosphate

As we presented in section 6.2, an in-depth study of the dynamics of variation is needed for each commodity. Considering the wideness of the domain and the multitude of raw materials, we concluded that a pilot study had to be performed to assess the feasibility of this approach. We decided to choose phosphate for this pilot study.

Phosphate is not the most well-known commodity and its importance is usually underestimated. In this section, we present why we decided to choose phosphate and we transcribe the reasoning that led to this decision. First, we present the different steps of the reasoning and the context. Then, we present a preliminary study on the factors influencing the price of phosphate.

The choice

As part of our work to anticipate global issues associated with commodities, we have been led to propose an analysis of commodity prices. First, the commoprice tool [6] was used to model the prices of some commodities. This tool allowed us to create monthly indices of some commodities in a sectorial approach.

This first approach revealed a first trend of raw material prices (wheat, coal, gold, iron ore, oil, phosphates, platinum, etc.) towards the rise. This allowed us to identify the complexity of the field, present different price quotations (currency) depending on the unit of measurement used, the state of processing of the material, the delivery time, and the delivery conditions including free on rail (f.o.r.); freeon board (f.o.b.); and cost insurance freight (c.i.f.). (Radetzki and Wårell, 2016) argues that price differences are greater for products with low values per unit weight and long transport distances.

Since the commoprice tool does not allow access to data prior to 1999 and the units of measurement may change depending on the commodity, the use of this tool for further work was not considered. In addition, the statistical time series most commonly used in the literature, as well as the price indices of the main commodities, are published by international organizations. Among others, the monthly International Financial Statistics by the International Monetary Fund is accessible on the Internet for free at this address[7]. Other publications more specialized in commodities also

[6] `https://commoprices.com/fr`

[7] www.imf.org/external/np/res/com mod/index.aspx

Table 6.2: Location of SCALIAN's offices and the sectors of activity of the Group's clients

Country	Main business sector
France	Aeronautics
Germany	Transports
Spain	Telecommunications and IT
Great Britain	Industry, Energy & Health
Belgium	Public sector and Services
United States	Banks, finance & Insurance
Canada	Spatial & Defense

publish monthly reviews, such as the Metal Bulletin in the United Kingdom. Thus, publications specific to the chosen commodity will be consulted for the remainder of the study.

Our first intuition was to focus on metals for the case study, as Scalian is historically tied to aeronautics, particularly in France and in Toulouse. However, the Covid-19 crisis has strongly impacted the aeronautics and transportation sectors, the main sectors of Scalian's current customers. Indeed, this has brought up the need and the will to diversify the customer portfolio, while applying the business know-how in other sectors. The chosen raw material had therefore to take into account strategic issues for Scalian's economic development in France and internationally, especially in Europe and North America. The table 6.2 shows the locations of Scalian's offices, as well as the main sectors of activity of the Group's current customers.

The European Commission launched the European Raw Materials Initiative in 2008 as part of an integrated strategy approach to establish measures to better address the challenges of securing and improving access to and management of raw materials. It publishes a list of critical non-energy commodities for the European economy, reviewed every three years, presented in Figure 6.5. The methodology for identifying critical materials takes into account two main components: economic importance and supply risk. The first is calculated based on the importance of a given material in the EU for end-use applications and on the performance of its substitutes in those applications and the second is based on factors that measure the risk of disruption in the supply of a given material.

This list was used as a starting point to choose a raw material that is strategic for Scalian and its societal environment in the Occitanie region in France, but also internationally. In this study, we want to use a raw material of scientific interest, which presents a complexity of forecasting its price intrinsic to its supply chain and which has a worldwide impact. To avoid problems of consumer behavior in view of a substitute and complementary products as mentioned in the previous iteration, the

2020 Critical Raw Materials (30)			
Antimony	Fluorspar	Magnesium	Silicon Metal
Baryte	Gallium	Natural Graphite	Tantalum
Bauxite	Germanium	Natural Rubber	Titanium
Beryllium	Hafnium	Niobium	Vanadium
Bismuth	HREEs	PGMs	Tungsten
Borates	Indium	Phosphate rock	Strontium
Cobalt	Lithium	Phosphorus	
Coking Coal	LREEs	Scandium	

Figure 6.5: 2020 Critical raw Materials. (Blengini et al., 2020)

raw material chosen for the development of the rest of the study should not have these variables to monitor.

In this way, after consultation with the business and technical experts, phosphate (Phosphorus and Phosphate Rock) was chosen for the continuation of the study since it meets the criteria established for the development of the study of weak signals. Indeed, this material is identified as critical for Europe, it is part of the six main elements of life (with oxygen, hydrogen, potassium, nitrogen, and carbon). It is mainly used in the food and defense industries, sectors of activity that still need to be strategically developed as part of the diversification of Scalian's portfolio.

Factors impacting the price of the selected raw material: phosphate, fertilizers and food commodities

(Mensah, 2003) was the first article consulted for the case of phosphate. It analyzed the price variations of phosphate on the international market, by making a statistical assessment (Labys et al., 1999) of prices from the 1950s to the 2000s. The study attributes the price variations to the dynamics of the production-demand variables, variations in available stock by the producing countries and the strategy of the players, notably Morocco (through the OCP company) and the United States. The analysis does not take into account the post-2000 period, nor the effects of speculation and the Internet. A diagram of the main factors identified is presented in Figure 6.6.

(Heckenmüller et al., 2014) reviews the literature and recent data on phosphorus availability and discusses the main determinants of world phosphate market prices. It analyzes the main importers and exporters (2011 data), showing China as a major exporter in the Asian market. It demonstrates that past fluctuations in phosphate and phosphate fertilizer prices do not reflect the physical depletion of phosphate rock, but rather are attributable to many other demand and supply factors. The main factors identified are in its majority around production variables but the authors also indicate the link between phosphorus prices and agricultural commodity prices.

(Cordell and White, 2015) propose a comprehensive set of indicators of phosphorus vulnerability and security at the global and national levels in relation to the principles

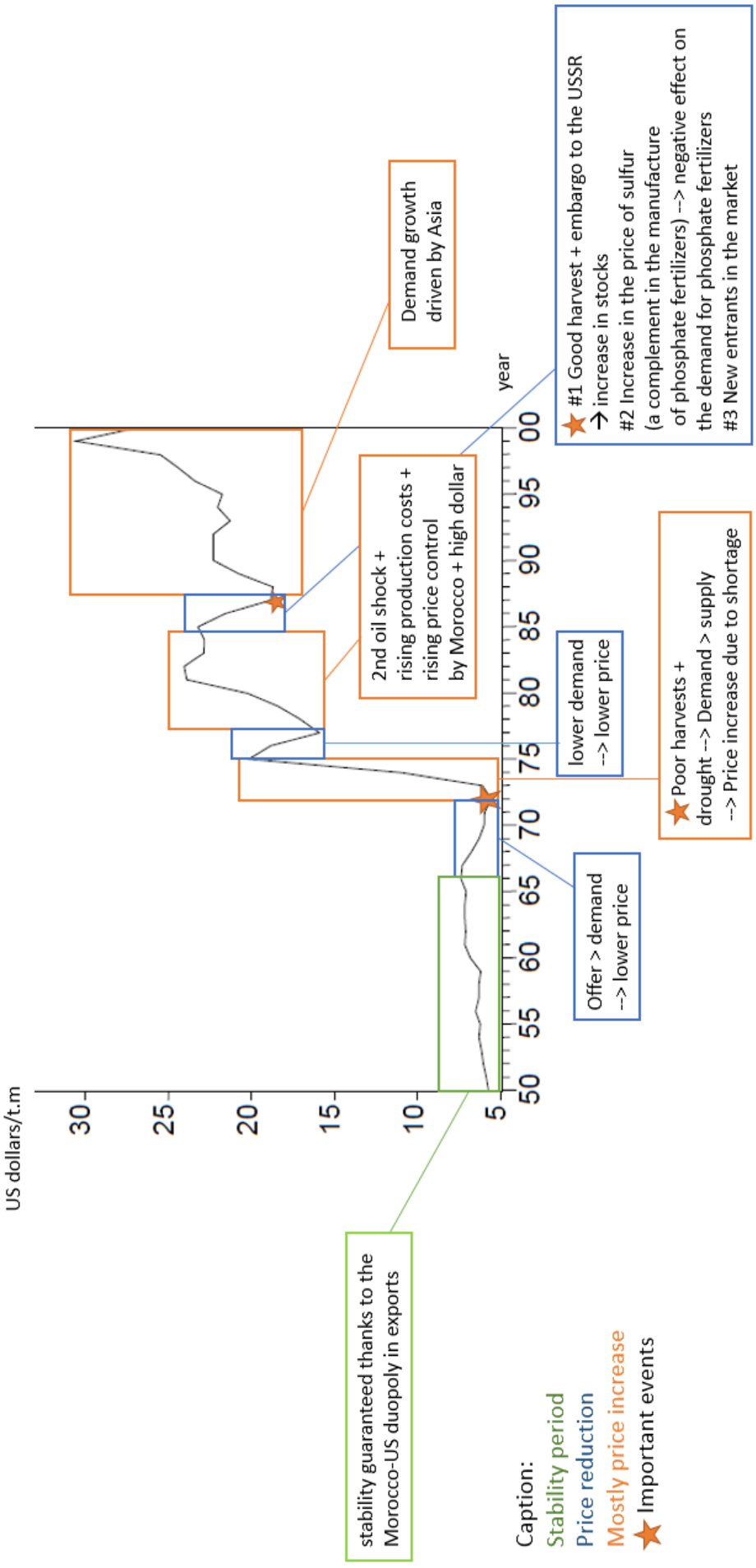

Figure 6.6: Evolution of the price of raw phosphate in US Dollars/t.m - Years 1950 to 2000. Adapted from (Mensah, 2003)

of sustainable development associated with food security. In order to determine which indicators can impact the price of phosphorus, the indicators were chosen in relation to their effect on the supply and/or demand of phosphate and therefore its price. They consider risks related to political instability in producing countries. (Cordell et al., 2015) also assessed the risks of the multi-stakeholder supply chain and stakeholder interaction. Many of the risks identified are upstream of the supply chain, i.e., at the phosphorus mining activity, impacting numerous stakeholders at all levels of the chain; but there are also downstream risks, which impact the actors at the beginning of the chain, thus confirming the interdependence of the stages and the "system feedback" effects, also mentioned by (Heckenmüller et al., 2014) and illustrated in Figure 6.7.

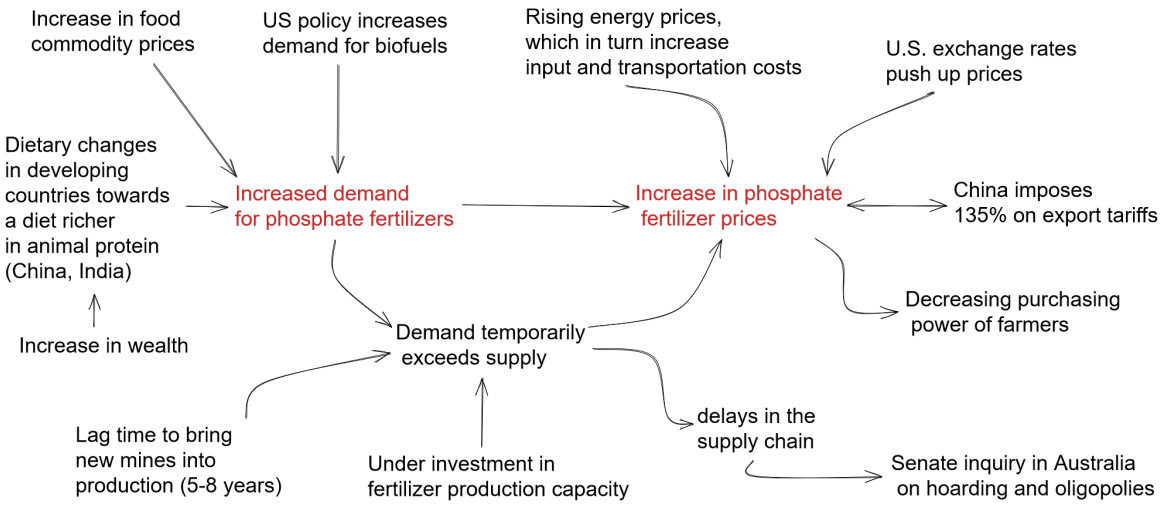

Figure 6.7: Schematic of the interdependence of steps and feedback effects of systems for phosphate. Adapted from (Cordell et al., 2015)

(Mew et al., 2018) has published a study based on a transdisciplinary approach, with the publication of a review of the state of the art on several topics related to the complexity of the phosphorus supply chain. On the economic side, the authors analyze the market dynamics and the price negotiation model of phosphate rock and phosphate fertilizers, which is established in private agreements between buyers and sellers. Therefore, they defend that the prices agreed in this way follow the effects of supply and demand in the economic context, with actors interacting on the feedback control mechanism (Wellmer and Dalheimer, 2012).

6.3.5 Partial conclusion

In this section, we introduced the definition of raw materials and some insights into what can influence the commodities' stock prices. We also showed that social media analysis can help to understand stock market variations. Finally, we presented some elements about how and why we chose to focus on phosphate. Even if some work identified some elements that influence the price of phosphate, it is not clear from the

literature what sources and what types of events to monitor to anticipate the variations of the price. Hence, in the next section, we apply our method to determine what events to monitor to anticipate raw materials price variations and apply our findings to the case of phosphate.

6.4 Results of the study

As mentioned previously, in the context of this study, phosphate was chosen as the raw material for the development of the pilot case of the event detection system on social networks. For terminological clarification purposes, the difference between phosphorus (P in the general sense and P4 with reference to commodity) and phosphates results from their chemical composition. They represent the chemical element and its commodity, respectively, whereas phosphates are compounds when phosphorus is bound to oxygen and other mineral elements (De Ridder et al., 2012). To clarify, phosphate rock in the form of P2O5 (i.e. phosphorus pentoxide) is a practical and standard measure of the phosphorus content of any product (Mew et al., 2018) and therefore will be privileged for the rest of the study.

Beyond its essential character for life, it is non-substitutable and exhaustible. Together with nitrogen (N) and potassium (K), they form the bricks of modern fertilizers (NPK), making them one of the key raw materials for food security. In fact, phosphate is mainly used in the production of fertilizers, in animal feed and in food additives, and a small fraction in industrial processes. The phosphate supply chain begins with the mining of phosphate rock. Then, treatments are applied according to the desired use (Blengini et al., 2020). The simplified supply chain of rock phosphate and element phosphorus is illustrated in figure 6.8. Likewise, the main by-products of each step of the process, as well as the distribution of the main uses in the European Union are presented. The main usages for the phosphate rock are to produce fertilizers for agriculture and as an input for animal feed. Despite its non-substitutable character in agricultural applications, the same statement is not applicable in industrial uses (Heckenmüller et al., 2014). Therefore, this type of application will not be considered for the rest of the study, nor the recycling process, making the phosphates food supply the subject of this research.

Thus, the phosphates food supply chain comprises the sectors and processes related to mining, phosphate rock/phosphorus processing and trade, fertilizers production and trade, agriculture application of fertilizers in crops and pastures, food production, processing, and distribution, and the final consumption. Hence, there is an industry's trend toward vertical integration which was highlighted in (Van Kauwenbergh, 2010).

In the literature review to identify factors affecting raw materials prices, the main findings were that price variations were frequently caused by a perturbation in the

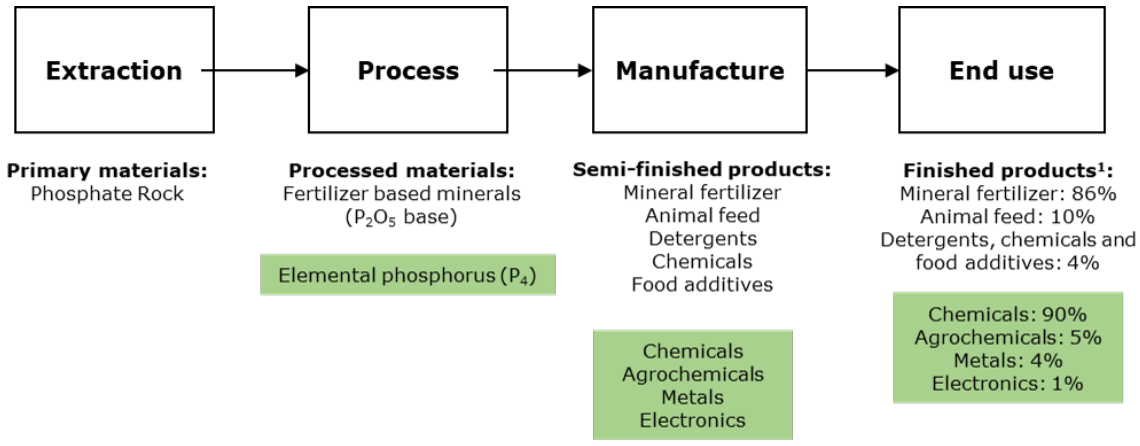

Figure 6.8: The simplified phosphate supply chain. Adapted from (Blengini et al., 2020)

supply-demand balance. This hypothesis was also used to explain major historical price changes for Phosphates (Cordell et al., 2015, Heckenmüller et al., 2014, Mensah, 2003, Mew et al., 2018). Given the goal of identifying the events impacting the price of phosphate, it is necessary to understand the dynamics of supply and demand of the market. In this way, the world reserves and the main phosphate rock producing countries have been identified and they are presented in the tables 6.3 and 6.4 and illustrated in figures 6.9 and 6.10. The data presented in the tables below are until 2018 because it was the latest publication when this study was conducted.

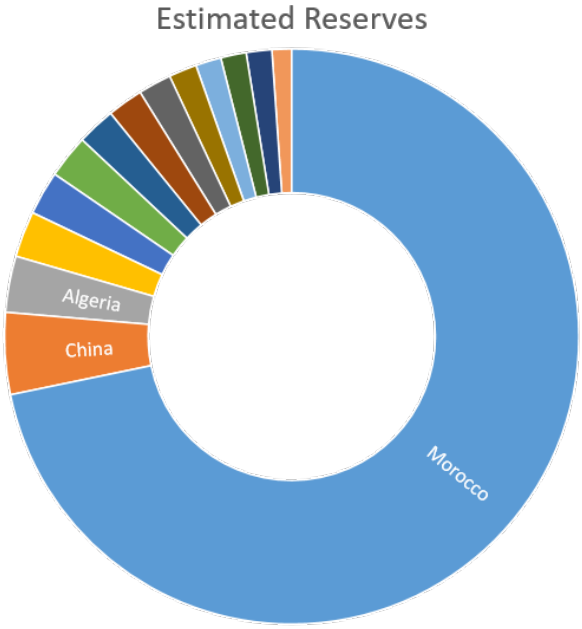

Figure 6.9: Distribution of world phosphate rock reserves.

Table 6.3: Phosphate rock world reserves. Source: (Jasinski)

Country	Estimated Reserves (tonnes)	Participation percentage
Morocco and Western Sahara	50 000 000	71.8
China	3 200 000	4.6
Algeria	2 200 000	3.2
Syria	1 800 000	2.6
Russia, Peru, Kazakhstan, Tunisia, Uzbekistan, Israel, Senegal, India, Mexico and Togo*	1 713 000	2.5
Brazil	1 700 000	2.4
South Africa	1 500 000	2.2
Saudi Arabia	1 400 000	2.0
Egypt	1 300 000	1.9
Australia	1 100 000	1.6
United States	1 000 000	1.4
Finland	1 000 000	1.4
Jordan	1 000 000	1.4
Other Countries	770 000	1.1
World total (rounded)	70 000 000	

It is in Morocco where most of the world's phosphate rock reserves are found (71.8%) and China takes the place of the world's largest producer (50.7%). There is a high concentration of the market, as five countries (China, Morocco, USA, Russia, and Jordan) produced around 80% of the phosphates in the world. This highlights a particular dynamic of the world market, which corresponds to the import and export balance of countries, their characterization as producing, exporting, or exclusively importing countries. Therefore, an analysis of the trade balance between the countries was carried out on the World Bank data for the main phosphate products including Phosphorus (P4), Phosphate rock (P2O5). The results are shown in figures 6.11 and 6.12. These data are accessible on the World Bank's website [8].

They confirm the market concentration and dependence of many countries on imports, of which about 91% depend completely on the import of phosphate rock and/or phosphorus to meet their internal demand. Since Phosphorus (P4) is obtained after processing, the main exporting countries are not the same as for the Phosphate Rock. This highlights the dependence of a large amount of the globe that relies on few producers and the concerns on phosphorus availability for food security (Cordell and White, 2015). On the one hand, monitoring events that could disturb production in the producing countries is then crucial for importers to anticipate their response to eventual supply disruptions. On the other hand, if there is a non-expected demand peak, producers should be able to anticipate and invest in production expansion, since

[8] `https://wits.worldbank.org/`

Table 6.4: World Production. Source: (Jasinski)

Country	Average (2014-2018) in thousand metric tonnes	Participation percentage
China	38 960	50.6
Morocco	9 074	11.8
United States	7 486	9.7
Russia	4 766	6.2
Jordan	2 572	3.3
Brazil	1 996	2.6
Saudi Arabia	1 571	2.0
Egypt	1 397	1.8
Peru	1 123	1.5
Israel	1 116	1.5
Tunisia	1 090	1.4
Vietnam	892	1.2
Australia	758	1.0
South Africa, Mexico, Senegal, India, Algeria, Finland, Togo, Turkey, Kasakhstan, Syria, Uzbekistan, Iran, Nauru*	4039	5.3
Other countries	86	0.1
World total	76926,05	100

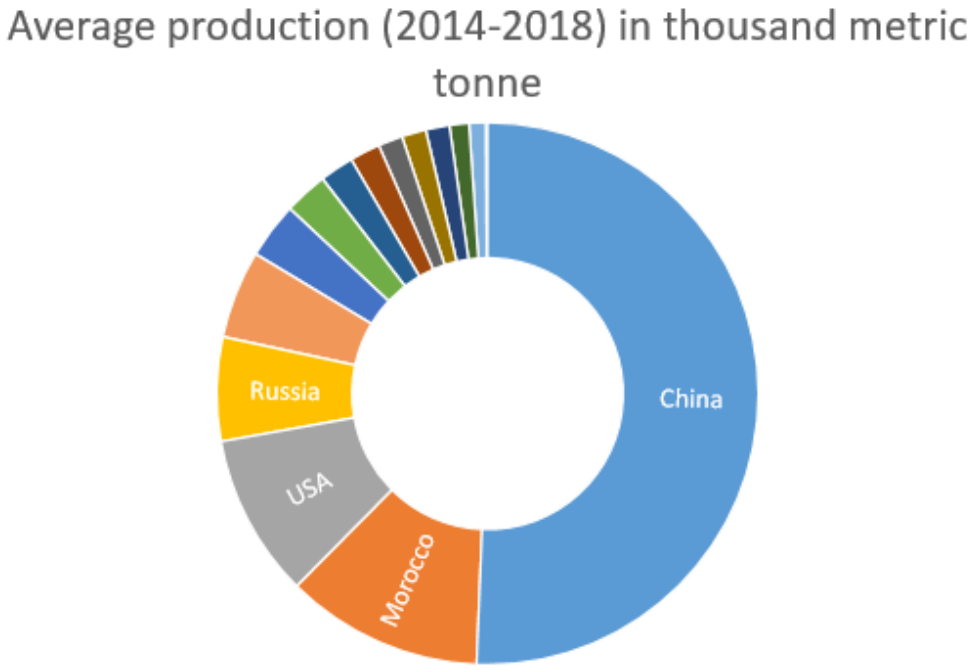

Figure 6.10: Distribution of world phosphate rock production (2014-2018).

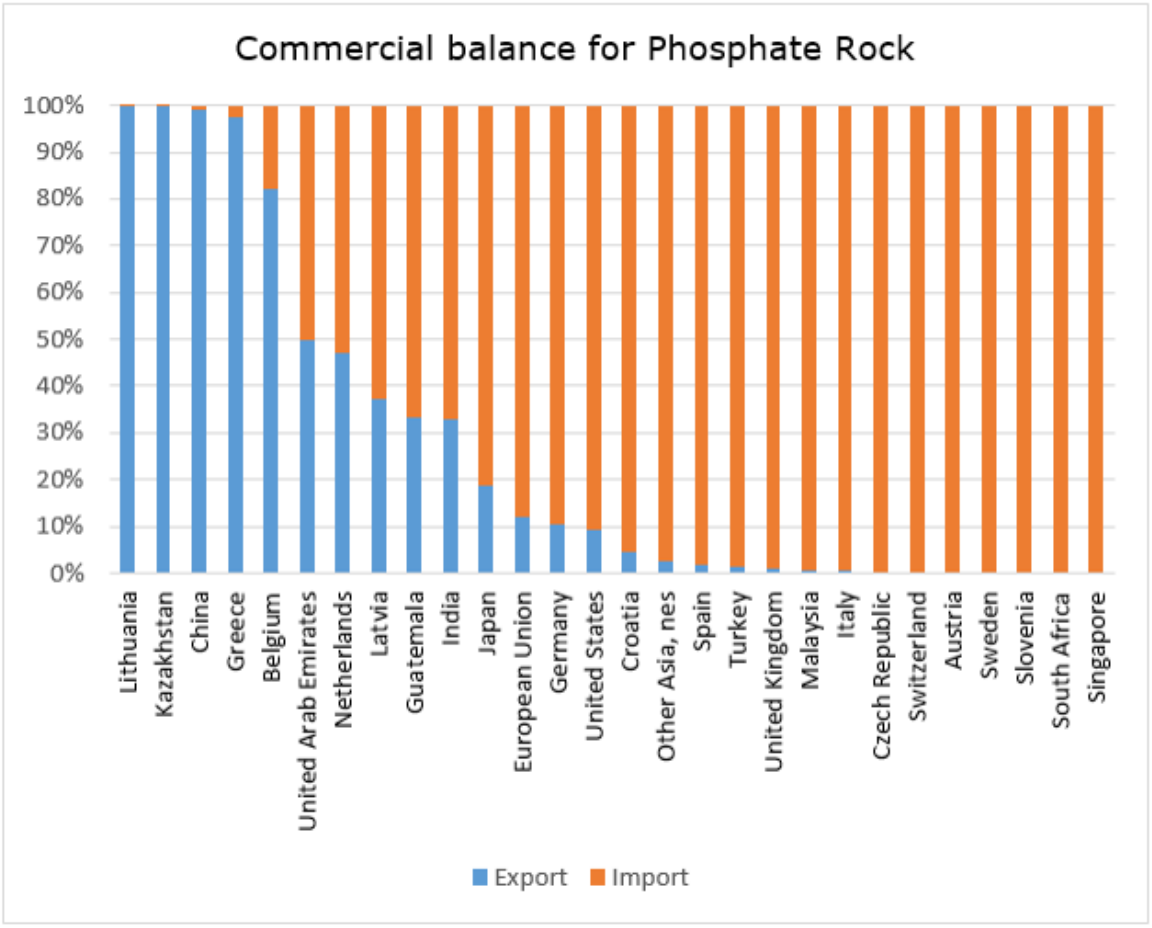

Figure 6.11: Phosphorus commercial balance. Source: World Bank Data, Accessed on 14/12/2020

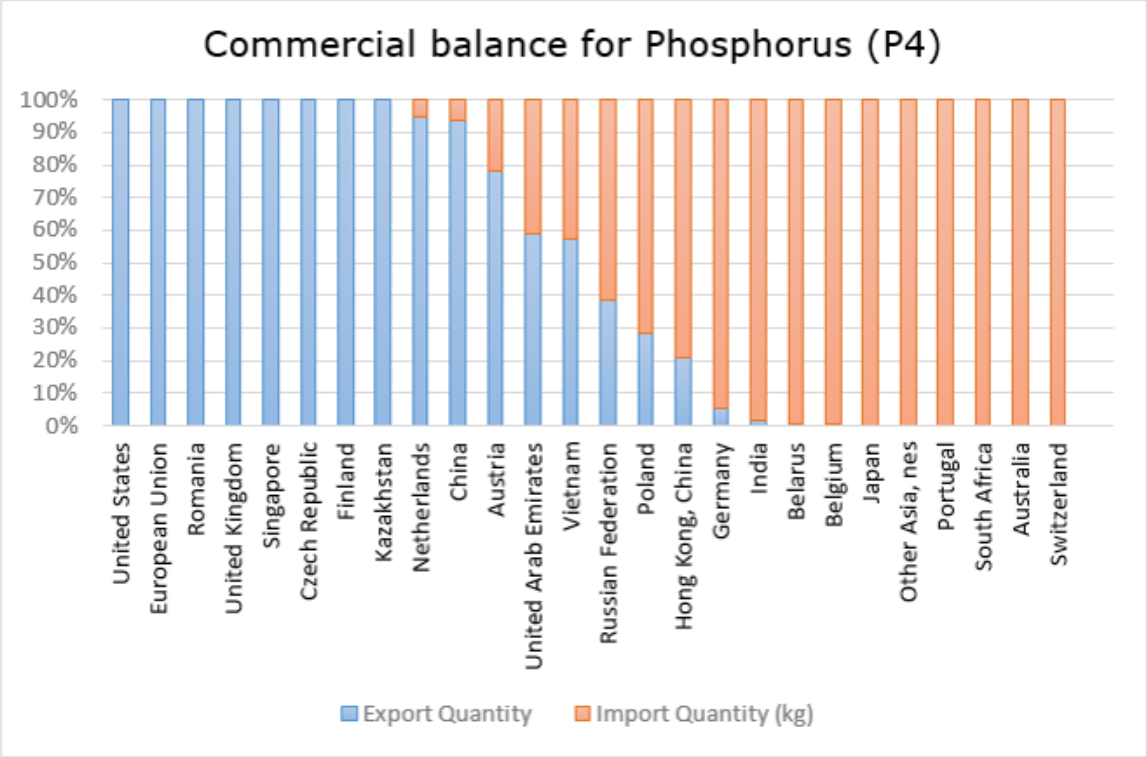

Figure 6.12: Phosphorus commercial balance. Source: World Bank Data, Accessed on 14/12/2020

the time gap between the investment decision and the operational plant can be 3 to 5 years (Weber et al., 2014).

From there, the identification of geographic areas to be supervised as well as the major players in the global market was initiated, in view of the supply chain previously presented. As mentioned before, these actors represent the majority of transactions in the field for the food supply chain and can contribute in different ways to market balance. They are presented in the form of a simplified stakeholder matrix in the table presented in the appendix A, which will serve as a basis for the construction of event detection filters, with the aim of defining the types of documents that are published and the types of events that we want to search according to the players supervised.

An initial sample of 135 main players was identified and divided by sector, role, geographical zone, and their side in the economic balance (demand, supply, or context) for the phosphates use case. For companies from the mining sector and in the production of fertilizers, as they are upstream in the value chain, the mine location was used; the downstream, for the agribusiness companies, their headquarters' location was considered. Transversal actors from the supply chain, such as business associations, non-governmental organizations, institutions, the academy, and specialized international research groups. They were considered as context factors, providing information for the macroeconomics condition of the market.

As (Van Kauwenbergh, 2010) highlighted the similarities in the price behavior

between the fertilizers and the phosphates, as well as the industry trend toward verticalisation, their intrinsic relationship should be exploited. Hence, members from the International Fertilizers Association (IFA) were also identified for further monitoring. They represent more than 400 institutions from 72 different countries.

Finally, once the main players and the geographic areas have been identified, the chronological analysis of the phosphate price was carried out through a review of the state of the art on the historical peaks of phosphorus. To create a typology of events that have impacted the price and could impact it again in the future, they are categorized as affecting supply, demand and/or the entire system, related to the economic context. The events' impact period (short or long-term variations) is also highlighted. Factors identified as risks to food security by (Cordell et al., 2015) such as political instability were also considered with regard to its influence on the availability of supply. The figure 6.13 presents the outcomes of this analysis, inspired by the work of (Rezitis and Sassi, 2013) for commodity food prices.

On the one hand, phosphates supply is mainly influenced by factors related to production disruptions due to higher production costs (ex: energy and sulfuric acid). In addition, the high production concentration in politically unstable countries put in evidence the need to monitor political events, as well as exportation policies. On the other hand, short-term triggers for a demand peak could be related to production shortfalls due to extreme weather conditions. Biofuels politics is also an interesting point to be looked to since it can burst the fertilizer demand to grow biofuel crops (Cordell et al., 2015).

Events in market and macroeconomics conditions are those that characterize the economic period and context. The main findings in the literature were related to the USD exchange rate since all contracts are negotiated in this currency; there is a growing influence of speculation and the financialization of the commodities market in the literature. (Anani, 2019) defends that this movement started by the Commodity Future Modernization Act (CFMA) when stock investors from the USA were allowed to place their assets in raw materials. As in (Kaldor, 1976), the economic function of speculation is to mitigate price variations due to changes in supply or demand, this mechanism influences the commodities and also the phosphates market.

Different studies put in evidence the risks dynamics of the phosphates multi-stakeholder supply chain. (Cordell and White, 2015) identified different risks throughout the food supply chain and its link to phosphates availability, and how they are transferable to other layers. It reveals the interdependence of the steps and actors of the supply chain, which creates a systems feedback effect that was also featured in (Wellmer and Dalheimer, 2012) and (Heckenmüller et al., 2014) and should be taken into consideration for the event detection. Thus, this is also evidentiated by the similar behavior and trend between food, fertilizers and phosphates prices previously

Potential triggers	Economic rationale	Time frame	Reference
	Supply factors		
Exportation politics	Some of the exporting countries introduce restrictive trade policies aimed at isolating their economies and controlling the market. Example: China in 2008	Short-term	Cordell and White, 2015
Low investivement in R&D and slower infrastrature response	Limitation of production capacities: the gap between the investment decision and the actual production is about 3 to 5 years	Long-term	Weber et Steiner, 2013 ; Heckenmüller et al. 2014
Political instability in producing countries	Examples: Tunisia during the Arab Spring and in Syria today due to the ongoing civil war (de Ridder et al. 2012; Taib 2013)	Short-term	Heckenmüller,et al. 2014 ; Cordell and White, 2015 ; Cordell et al, 2015;
Higher production costs	Energy prices, labor costs, costs of chemical reactants (e.g. water, price of sulfur and sulfuric acid)	Medium-term	Mew et al., 2018
	Demand factors		
Weather and crop quality	Agriculture production shortfalls due to adverse weather conditions lead to lower global food supply and higher fertilizers demand	Short-term	Mensah, 2003 ; IFA, 2011
Emerging Economies	Increased income in the BRICs and more urban population is changing food habits pushing demand for higher-value products	Long-term	Cordell and White, 2015 ; Cordell et al, 2015
Biofuels productions and politics	Increased demand for crops used as inputs in the production of biofuels;	Long-term	Cordell et al, 2015
Food commodity prices	Encourage farmers to increase their crop yields by applying more fertilizers, including phosphate fertilizers	Short-term	IFA 2011 ; Huchet-Bourdon, M. 2011
Importation politics	Protectices, import tax to promote the domestic economy	Short-term	Cordell et al, 2015
	Market and macroeconomic conditions		
USD change rate	The purchasing power of countries varies according to its currency	Short-term	Huchet-Bourdon, M. 2011 ; Cordell et al, 2015
Speculation and Financialization	Reference price on the stock markets, which influences production and consumption decisions.	Short-term	Cordell and White, 2015; Anani, 2019
Energy and fertilizers price	Inputs for agriculture: increase in production costs; increased production of biofuels	Medium-term	Heckenmüller,et al. 2014 ; IFA, 2011 ; Huchet-Bourdon, M. 2011
Food commodity prices	The price of phosphate rock follows food and agricultural prices very closely, although they lag slightly behind (a month or two)	Short-term	Van Kauwen-bergh 2010 ; Saywell, 2013 ; Heckenmüller,et al. 2014

Figure 6.13: Events that influence Phosphates price. Adapted from (Rezitis and Sassi, 2013)

mentioned.

Major long-term trends, such as changes in consumers' income and therefore in their diets should be less exploited for further analysis since short-term events are privileged. The risk of resource depletion will not be considered either, since it has been proved as having no real connection with price variations (Heckenmüller et al., 2014) and due to the differences between existing current estimations and technological developments for optimizing industrial processes. The growing mention of the influence of stock market financial speculation on commodity prices will be further investigated for the remainder of the study.

In sum, the main event types to be supervised will be related to the countries' commercial politics (ex: export or import taxes, biofuels), political events (war, demonstration, riots), weather conditions in major crop areas, the food commodities market, as well as the sulfuric acid market; major financial indicators, such as the USD rate and the financial markets. These events affect the supply or the demand of phosphates and could then influence their prices. They were identified after a literature review on phosphates, but this approach could be extended to other commodities.

Most studies on raw materials analyze historical prices within the framework of an economic analysis of raw materials' volatility (Huchet-Bourdon, 2011), the challenges of sustainable development (Anani, 2019, Cordell et al., 2015), as well as the issue of price in relation to the resources' depletion (Henckens et al., 2016). Events have been identified as being the causes of price variations through statistical correlation and causality tests, notably on the effects of the price of fertilizers, crude oil, agricultural commodities, and the USD exchange rate (Huchet-Bourdon, 2011). Nevertheless, real-time event detection from social networks and news media for commodity price prediction has not yet been the subject of any referenced study.

Open points in the literature include the impact of biofuels (ie: bioethanol and biodiesel) and speculation. For the first, despite the difficulty of comparison between existing studies due to the different methodologies adopted, most authors agree on the fact that the expansion of the consumption of biofuels has an upward impact on the price of food (Rezitis and Sassi, 2013). For the latter, there is an inherent complexity as pointed by (Palazzi et al., 2020) that even sophisticated models are not capable of capturing the causal effects of speculation and the dynamics of the market.

For future works, it should be privileged to search for these events within the players previously identified, considering their main communication channels, but also the environment they are in. This could be particularly interesting in producing countries for monitoring eventual political events that could disturb supply, as mentioned by (Cordell et al., 2015). For those countries relying exclusively on importations, it could be interesting to monitor their local weather conditions and consequent crop quality to be aware of eventually demand peaks and anticipate a supply chain responsiveness. For

industrials from all sectors, additional events could be detected throughout the actors' Twitter accounts (when applicable) and monitoring the main institutional publications on the subject. For example, following the latest developments in infrastructure that could increase capacity or improve recycling methods.

Further research efforts will be dedicated to the quantification of these events' impact on the phosphates prices, to provide a trend to the event and facilitate decision-making. Specific investigation on the speculation should also be addressed, since its inherent complexity and causality effects in multiple markets. Recent technology developments such as cryptocurrencies could also be a subject for further research as it increases the possibilities of transactions.

6.4.1 Experiment on Twitter's data stream

Detail of the experiment

To illustrate our proposition, we experimented with Twitter's data stream, calibrating the filters using the results of the study presented in the previous section. We filtered the stream using simplistic filters, i.e. based only on static keywords derived from the tables presented in appendix A. In particular, we used the keywords summarized in B.

We retrieved all the tweets in the English language containing at least one of these keywords between the 17th of April 2022, 18:00 (UTC) to the 19th of April 2022, 11:00 (UTC). We retrieved 99 616 tweets and applied to them the same filtering rules as mentioned in chapter 5, to obtain a final number of 66 585 tweets. We applied EDS to these tweets. A sample of these results are presented in figures 6.14 to 6.17. More examples are provided in appendix B.

Discussion of the results

Several events are very informative. Most of them discuss events related to China, as can be expected from the list of keywords established, and the relative popularity of each word. We did not see any mention of the word 'phosphate' during our review of the events. An interesting observation is that very few event clusters are spam, except maybe the baseball-related ones presented in appendix B, but they are considered interesting in some datasets. However, we see that filtering using only keywords as a white list is not sufficient. Indeed, figure 6.17 shows that using "Lindt" as a keyword is unlikely to provide insights about the results or the activity of the company, but much more about chocolate-related discussions. The same observation can be made for the keyword "Mars", where most events are about Bruno Mars or the planet Mars. Several methods can be considered to gather more representative data, such as the method employed to constitute the Events2018 dataset (Mazoyer et al., 2018).

Sample tweets	‘Chinas Economic Data Hints at Cost of Zero CovidStrategy’ ‘You cant believe the economic stats out of China The sad thing is you cant believe the economic stats comin’ ‘Chinas Economic Data Hints at Cost of Zero Covid Strategy via NYT’ ‘CHINA STATS BUREAU SPOKESMAN WE WILL INCREASE OUR ASSISTANCE TO INDUSTRIES AFFECTED BY COVID’ ‘CHINA STATS BUREAU SPOKESMAN CHINA WILL BE ABLE TO CONTAIN COVID AND REDUCE ITS ECONOMIC IMPACT’
Named entities	[('Chinas Economic Data Hints', 26), ('Zero', 16), ('China', 6), ('Keith Bradsher', 2), ('zero', 1), ('Chinas Financial Facts Hints at Expense', 1), ('NYT', 1), ('NBC', 1), ('CHINA STATS BUREAU SPOKESMAN WE', 1), ('Gordon Chang', 1)]

Figure 6.14: China published data about their economics results. These data show that the zero covid policy have an impact on their economy.

Sample tweets	‘US Treasury Secretary Yellen Will urge IMF World Bank to increase econ pressure on Russia’ ‘World Bank slashes global growth forecast to 3 2 from 4 1 citing Ukrainewar’ ‘World Bank slashes global growth forecast to 3 2 from 4 1 citing Ukrainewar ‘ ‘World Bank slashes global growth forecast to 3 2 from 4 1 citing Ukrainewar’ ‘World Bank slashes global growth forecast to 3 2 from 4 1 citing Ukrainewar’
Named entities	[('World Bank', 45), ('3 2', 29), ('4 1', 27), ('Ukraine', 26), ('Russia', 13), ('Ukrainewar', 11), ('2022', 8), ('US', 7), ('Treasury', 7), ('EU', 5)]

Figure 6.15: World bank reacts to the Ukrainian war

Sample tweets	‘USGS reports a M1 01 earthquake 12km NNE of Coso Junction CA on 4 18 22 14 27 52 UTC earthquake’ ‘USGS reports a M 0 44 earthquake 20km ESE of Little Lake CA on 4 18 22 14 24 55 UTC earthquake’ ‘USGS reports a M0 95 earthquake 1km NNE of The Geysers CA on 4 18 22 19 13 23 UTC earthquake’ ‘USGS reports a M1 3 earthquake 3 km WNW of Walker California on 4 18 22 19 33 22 UTC earthquake’ ‘1 30 magnitude earthquake occurred at Golden Gate Rd Coleville CA 96107 United States on 2022 04 18 19 33 22’
Named entities	[('UTC', 19), ('USGS', 18), ('4 18 22', 17), ('14', 3), ('ESE', 3), ('United States', 3), ('Denali National Park Alaska', 2), ('Little Lake CA', 2), ('Denali AK United States', 2), ('M1 3', 2)]

Figure 6.16: Multiple event clusters are about USGS reporting earthquakes. It is not exactly the kind of content we expected about USGS, but it can be important.

Sample tweets	‘What is the best chocolate and why is it Lindt’ ‘I love these Lindt rabbits theyre a classic in my house around most holidays’ ‘Happy Easter and happy birthday to Lindt Bunny xx’ ‘Good baby I know you love Lindt’ ‘Happy chocolate egg day to those who worship the gods of Lindt Thorntons and Cadburys’
Named entities	[('Lindt', 6), ('Lindt Bunny', 1), ('Lindt Thorntons', 1), ('a month ago', 1), ('Bunny', 1), ('Mint Lindt', 1), ('Lindt Easter', 1), ('tomorrow', 1)]

Figure 6.17: People are discussing chocolate during Easter.

Overall, we think the results are promising. The event clusters are coherent and their content allows an easy understanding of the events discussed. To show the results, we directly extracted 5 tweets out of 10 random sampled tweets of the event clusters and the most frequent named entities. As we can see, events are easily understandable using these summaries.

However, they are still a lot of mundane conversations and the focalisation to potentially impacting events is currently not achieved. Thus, it highlights the need to more carefully engineered filtering rules to retrieve relevant tweets, to reduce the number of mundane conversation clusters. The performance and the quality of the produced clusters of EDS are nonetheless up to our expectations.

6.5 Conclusion

In this chapter, we investigated which factors influence raw materials price fluctuations through a literature review. We found that events affecting the commodities stocks are those linked with the demand and supply of the commodities and macroeconomic conditions. A case study on phosphate was conducted using the previous results. We investigated institutional sources such as the World Bank, USGS, and IFA to identify the main typology of impacting events (political, weather conditions, food supply chain, currencies variations, and petroleum prices) and determine private companies and geographical locations of interest.

Then, we applied these results during a primary study to illustrate our result and estimate the usability of our model. Future works will be dedicated to applying the results in a more developed way. For this, the tables of actors to be supervised, as well as the main categories of events affecting the supply or demand have to be considered. To do so, identifying relevant keywords as well as relevant sources, such as Twitter accounts is necessary. We also want to study the effects of the speculative derivatives market on the price of commodities to better understand this mechanism and its relationship to weak signals in social networks. Finally, the replicability and transposition of the process for identifying events impacting not only other raw materials but also other components of any supply chain will be studied in more depth.

Chapter 7

Conclusion

In this thesis manuscript, we presented and discussed the more relevant results of our research. First, we summarize these results and then present some of the perspectives opened by our work.

7.1 Summary of the thesis

In this thesis, we tackled the problem of event detection in data streams. In particular, we focused on real-time open-domain event detection on social network text data stream, meaning that we wanted to detect events without prior knowledge on them, whether of type, number, size, or duration. To this end, we first presented some related work with a special emphasis on event detection on social network data streams and proposed an adaptation of a classical event detection framework that suits our needs. On top of this event detection framework, we designed and build an event detection system, EDS, that fulfills these requirements. To select and validate the different components of the system, we have carried out various experiments and research. First, we compared the performances of many text representation models and validated the performances of our system in a context of event-related documents clustering. Then, we proposed to combine lexical, semantic, and social network-specific representation models. We have shown that depending on the type of application, these combinations can be interesting. Then, we focused on the two last components of our event detection system, namely the event detection and the event summarization & tracking components. To evaluate properly these components, we proposed an evaluation method that suits classic datasets from the literature and improves the reproducibility of the results. We applied this evaluation method to evaluate the full event detection system and compare it to other systems of the literature. We concluded that EDS is competitive with these systems. Finally, we applied our work to the industrial context of this thesis, i.e. to supply chains and particularly those of raw materials. We proposed a framework that integrates EDS and synergizes it with a

business component, to involve supply chains managers and raw materials purchasers in the process, to ensure the robustness of the system. We also conducted a pilot study on phosphate, a critical commodity, to identify the impacting events that have to be detected and monitored to mitigate the disruptions they may cause.

In the next section, we propose different perspectives for future research offered by our work.

7.2 Perspectives for future research

In this section, we propose different ways to pursue our research at different time scales.

7.2.1 Short-term perspectives

A sentence model adapted to Twitter

As we highlighted in the related work, BERT-based models are really popular. Two models have been presented, TweetBERT (Qudar and Mago, 2020) and BERTweet (Nguyen et al., 2020). They are adapted to classical NLP tasks on Twitter such as POS tagging or NER. However, there is no model adapted to the embedding of tweets, such as S-BERT (Reimers and Gurevych, 2019) or USE (Cer et al., 2018) for classical sentences. We think the lack of a labeled dataset in which tweets are annotated as similar or dissimilar is the major reason for this. We believe there are two possibilities to answer this issue. First, annotating a dataset of tweets. Of course, it is really costly, so it might not be the optimal solution. Another solution is to reproduce an architecture similar to S-BERT, namely using siamese neural networks (Bromley et al., 1994), composed of two BERT-models fine tuned on tweets. Then, the siamese architecture could be tuned using the same datasets as S-BERT, even if it is not specialized in the representation of tweets. The results might be less optimal than results obtained using a dedicated dataset, but we believe it would still improve the performances.

Improvement of the performances of the system

Even if the performances of the system are decent, some optimizations are possible. One of the most time-consuming steps of the system is the similarity calculation during the document clustering step. A good way to improve the performance would be to parallelize this calculation. Another perspective of improvement would be to parallelize the document clustering step and the event detection step. Once the clusters are computed, they can be analyzed while other documents are clustered. This would also greatly improve the performance of the system in terms of computation time.

7.2.2 Medium-term perspectives

Improve decision making of the event detection module

We did not make a contribution to this step and used an approach from the literature. However, we believe that it has a lot of potential to improve the performance of the system. We think that exploiting the structure of the network to determine whether the community is discussing an event or not is a promising approach. Classifying the clusters as event-related or non event-related using the organization of the community is a potential method to consider. A technical solution to do so could be graph neural network (GNN) (Wu et al., 2020), which gained a lot of attention in the past years. The objective of these deep learning methods is to perform inference on data described by graphs. As we have seen in the previous sections, social networks can be represented as graphs, thus, several applications for GNN on social networks were explored, such as social recommendation (Fan et al., 2019) or fake news detection (Benamira et al., 2019). We believe similar work can be conducted for event detection on social media, using graph neural networks to classify graphs representing clusters of documents. However, some possible limitations of this type of approach are inference time and memory usage. Indeed, in a context such as event detection, a fast response time is needed and due to the size of the data, GNN might be too expensive to apply to this task.

Automatic identification of relevant sources

As we saw during this thesis, filtering the input documents, including choosing the right sources, is crucial for the performance of an event detection system. We dedicated a whole chapter of this thesis to establishing which events to supervise and what type of sources to monitor to find potential information about these events. However, this study is limited in the sense that it provides a certain number of domains, actor names, or keywords to supervise. However, it does not necessarily provide the important users of the social network, which are the main sources of information of the network. Thus, a method that could explore the network and identify the important users or topics to monitor from a set of keywords or a study similar to the one conducted in chapter 6 would be of particular interest.

Evaluation of the quality and veracity of the events

As we saw in chapter 5, some event detection systems such as MABED are ranking the events depending on their impact. EDS has no such process, which would be interesting. However, ranking the events based on their popularity may not be relevant for our application, where important events may not be popular. Thus, it is necessary to establish some metrics which would quantify the potential impact an event may have

on the supply chains we monitor. Some perspectives could be to evaluate the sentiment associated with the event, as it can have an influence on the stock market (Bollen et al., 2011), using sentiment analysis techniques, or analyzing emojis in the clusters (Guibon et al., 2016). Exploiting external sources such as DBpedia or WordNet (Hamdan et al., 2013) can be interesting to enhance the contextualization and assess the quality of an event, such as in (Morabia et al., 2019) or (Pandya et al., 2020). An interesting approach could be also to analyze the content and the structure (Chagheri et al., 2011) of the target documents of the links shared by the users. Indeed, considering that the tweets are lacking contexts, analyzing external links is important.

Another important aspect is the veracity of the event. It is clear that most of the information related online is at least not reliable, as some sources spread fake news, reinformation (Maigrot et al., 2016) or hoaxes (Maigrot et al., 2018). An event detection system is subject to the same issues if it has no system to differentiate a credible event from fake news.

Summarization and visualization of the events

Summarizing and visualizing a cluster composed of several text documents is no easy task, and is an active part of the research. In this thesis, we used a very simplistic way to summarize the events, i.e. representing them using the most frequents named entities of the documents constituting the events. A more sophisticated way to represent the event would be to present to the user a structured representation of the event, where the location, the main protagonist, and their interactions are represented. Such an approach is proposed in the literature, notably by the authors of (Zhou et al., 2017) and (Li et al., 2017). Another meaningful representation such as a summary generated using summarizer models like (Miller, 2019) could be interesting. However, we experimented with this solution and this summarizer seems to be not suitable for the summary of clusters of tweets and may need some adaptation to this specific type of document. Evaluating the quality of such summarizer is not an easy task (Ermakova et al., 2019) and requires adapted solutions.

7.2.3 Long-term perspectives

Event-based time series prediction

Scalian was interested in detecting events that are potentially impacting, to help their supply chain managers and raw material buyers in their daily decision making, but they were also interested in predicting the evolution of the stock market. We decided in this thesis to consider that predicting the stock market variations was a too ambitious objective and chose to integrate the experts in the process as much as possible to let them analyze the detected events and anticipate the variation they will cause.

However, some approaches to forecasting the stock market variations can be considered. An example could be to link the event detected by EDS and a predictive model which would give the future trends of the stock market depending on these events. A potentially interesting solution could be to use a General Adversarial Neural Networks (GAN) (Goodfellow et al., 2014) for Time series prediction. A possible framework is presented in Figure 7.1 The principle is the following: A GAN is composed of two major parts: the generator and the discriminator. The generator tries to mimic the actual data and the discriminator tries to identify fake data produced by the generator. We want to produce time-series estimations, so our idea is articulated as follows: the generator part of the GAN will produce time-series estimations taking events as input. The discriminator will be fed with two inputs, the actual time series, and the fake time-series, which are generated by the generator. The objective of the generator is to be able to produce time-series estimations that are really close to reality, to fool the discriminator. The discriminator's objective is to have a maximum accuracy in its task to differentiate between fake and real input. Since the final output we want is a time series estimation, our general objective is to have a generator as optimized as possible. The discriminator is only used in the training loop, to give feedback to the generator, and to train it to produce valuable output. To give hints about the future time series variations, the generator will take as input the events we have previously detected, which are supposed to carry information that influences these variations. By training it properly, the generator will be able to extract information from the events and the feedback of the discriminator. The feedback from the discriminator contains information about the time series, which is not directly available to the generator. Indeed, the final objective is to have a generator that can predict time series variations, by only exploiting the events we detect. A more detailed description of this approach is described in (Maître et al., 2020).

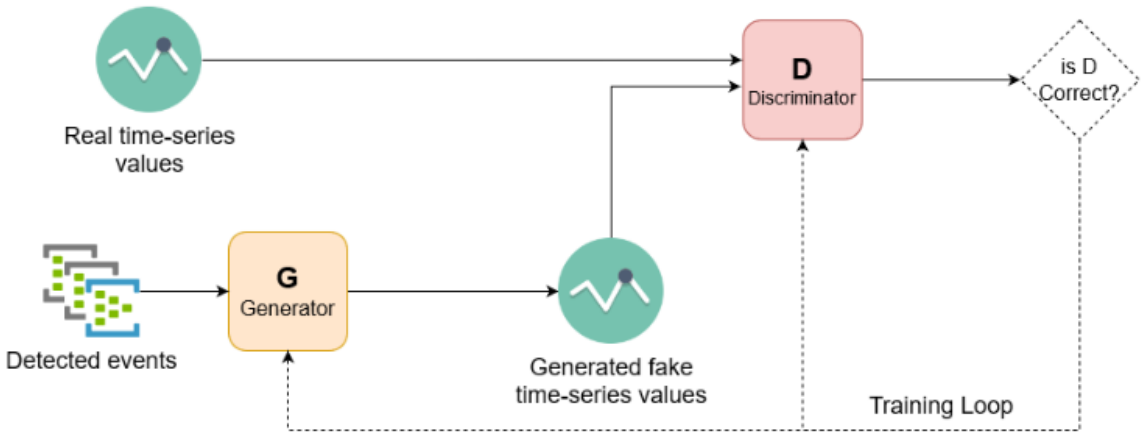

Figure 7.1: An example of GAN for time series prediction using detected events as input.

Bibliography

C. C. Aggarwal and C. Zhai. A survey of text clustering algorithms. In *Mining text data*, pages 77–128. Springer, 2012.

L. Aiello, G. Petkos, C. Martín Dancausa, D. Corney, S. Papadopoulos, R. Skraba, A. Goker, I. Kompatsiaris, and A. Jaimes. Sensing trending topics in twitter. *IEEE Transactions on Multimedia*, 15:1–1, 10 2013. doi: 10.1109/TMM.2013.2265080.

M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. Trippe, J. Gutierrez, and K. Kochut. A brief survey of text mining: Classification, clustering and extraction techniques. 07 2017.

J. Allan. *Topic detection and tracking: event-based information organization*, volume 12. Springer Science & Business Media, 2012.

J. Allan, J. G. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study final report. 1998.

J. Allan, V. Lavrenko, D. Malin, and R. Swan. Detections, bounds, and timelines: Umass and tdt-3. *Proceedings of Topic Detection and Tracking Workshop*, 11 2000.

E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486, 2009.

E. Anani. *Fluctuations des cours des matières premières: Enjeux de soutenabilité-Application à l'Afrique de l'Ouest*. PhD thesis, Université Paris-Saclay (ComUE), 2019.

M. Asgari-Chenaghlu, M.-R. Feizi-Derakhshi, L. farzinvash, M.-A. Balafar, and C. Motamed. Topicbert: A transformer transfer learning based memory-graph approach for multimodal streaming social media topic detection, 2020.

F. Atefeh and W. Khreich. A survey of techniques for event detection in twitter. *Computational Intelligence*, 31(1):132–164, 2015.

R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999. ISBN 0-201-39829-X.

H. Becker, M. Naaman, and L. Gravano. Learning similarity metrics for event identification in social media. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 291–300, 2010.

H. Becker, F. Chen, D. Iter, M. Naaman, and L. Gravano. Automatic identification and presentation of twitter content for planned events. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5, pages 655–656, 2011a.

H. Becker, M. Naaman, and L. Gravano. Beyond trending topics: Real-world event identification on twitter. volume 11, 01 2011b.

A. Benamira, B. Devillers, E. Lesot, A. K. Ray, M. Saadi, and F. D. Malliaros. Semi-supervised learning and graph neural networks for fake news detection. In *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 568–569. IEEE, 2019.

G. Blengini, C. Latunussa, U. Eynard, C. Matos, K. Georgitzikis, C. Pavel, S. Carrara, L. Mancini, M. Unguru, D. Blagoeva, F. Mathieux, and D. Pennington. Study on the eu's list of critical raw materials (2020) final report, 09 2020.

V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, P10008:1–12, Oct. 2008. doi: 10.1088/1742-5468/2008/10/P10008. URL `https://hal.archives-ouvertes.fr/hal-01146070`.

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8, 2011.

C. D. Boom, S. V. Canneyt, T. Demeester, and B. Dhoedt. Representation learning for very short texts using weighted word embedding aggregation. *CoRR*, abs/1607.00570, 2016.

S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

D. Boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *2010 43rd Hawaii international conference on system sciences*, pages 1–10. IEEE, 2010.

J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, pages 737–737, 1994.

M. Brzozowski and D. Romero. Who should i follow? recommending people in directed social networks. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5, 2011.

A. Burnie and E. Yilmaz. An analysis of the change in discussions on social media with bitcoin price. pages 889–892, 07 2019. ISBN 978-1-4503-6172-9. doi: 10.1145/3331184.3331304.

H. Cai, Y. Yang, X. Li, and Z. Huang. What are popular: Exploring twitter features for event detection, tracking and visualization. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, page 89–98, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450334594. doi: 10.1145/2733373.2806236. URL `https://doi.org/10.1145/2733373.2806236`.

C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684, 2011.

D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y. Sung, B. Strope, and R. Kurzweil. Universal sentence encoder. *CoRR*, abs/1803.11175, 2018. URL `http://arxiv.org/abs/1803.11175`.

S. Chagheri, S. CALABRETTO, C. Roussey, and C. Dumoulin. Document classification: Combining structure and content. In *13th International Conference on Entreprise Information Systems (ICEIS)*, pages p. – p., Beijing, China, June 2011. SciTePress. URL `https://hal.archives-ouvertes.fr/hal-00637665`.

P. Champagne. L'événement comme enjeu. In *Réseaux, volume 18, n°100. Communiquer à l'ère des réseaux*, 2000. doi: 10.3406/reso.2000.2231.

F. Chen and D. B. Neill. Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1166–1175, 2014.

H. Chen, P. De, Y. J. Hu, and B.-H. Hwang. Wisdom of crowds: The value of stock opinions transmitted through social media. *The Review of Financial Studies*, 27(5): 1367–1403, 2014.

A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1070.

D. Cordell and S. White. Tracking phosphorus security: indicators of phosphorus vulnerability in the global food system. *Food Security*, 7(2):337–350, 2015.

D. Cordell, A. Turner, and J. Chong. The hidden cost of phosphate fertilizers: mapping multi-stakeholder supply chain risks and impacts from mine to fork. *Global change, peace & security*, 27(3):323–343, 2015.

M. De Ridder, S. De Jong, J. Polchar, and S. Lingemann. Risks and opportunities in the global phosphate rock market: Robust strategies in times of uncertainty, 2012.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

C. Domeniconi and M. Al-Razgan. Weighted cluster ensembles: Methods and analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(4):1–40, 2009.

W. Dou, X. Wang, W. Ribarsky, and M. Zhou. Event detection in social media data. In *IEEE VisWeek Workshop on Interactive Visual Text Analytics-Task Driven Analytics of Social Media Content*, pages 971–980, 2012.

A. Edouard. *Event detection and analysis on short text messages*. Theses, Université Côte D'Azur, Oct. 2017. URL `https://hal.inria.fr/tel-01680769`.

A. Edouard, E. Cabrio, S. Tonelli, and N. Le-Thanh. Graph-based event extraction from Twitter. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 222–230, Varna, Bulgaria, Sept. 2017. INCOMA Ltd. doi: 10.26615/978-954-452-049-6_031. URL `https://doi.org/10.26615/978-954-452-049-6_031`.

L. Ermakova, J.-V. Cossu, and J. Mothe. A Survey on evaluation of summarization methods. *Information processing & management*, 56(5):1794–1814, Sept. 2019. doi: 10.1016/j.ipm.2019.04.001. URL `https://hal.univ-brest.fr/hal-02130700`.

W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pages 417–426, 2019.

Y. Fan, L. Heilig, and S. Voß. Supply chain risk management in the era of big data. In *Design, User Experience, and Usability: Design Discourse*, pages 283–294. Springer, 2015.

M. Fedoryszak, B. Frederick, V. Rajaram, and C. Zhong. Real-time event detection on social data streams. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery; Data Mining*, KDD '19, page 2774–2782, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330689. URL `https://doi.org/10.1145/3292500.3330689`.

X. Feng, L. Huang, D. Tang, H. Ji, B. Qin, and T. Liu. A language-independent neural network for event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 66–71, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2011. URL `https://www.aclweb.org/anthology/P16-2011`.

A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. *Acm transactions on knowledge discovery from data (tkdd)*, 1(1):4–es, 2007.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

R. Grishman, D. Westbrook, and A. Meyers. Nyu's english ace 2005 system description. *Proceedings of ACE 2005 Evaluation Workshop. Journal on Satisfiability*, 51, 01 2005.

G. Guibon, M. Ochs, and P. Bellot. From emojis to sentiment analysis. In *WACAI 2016*, 2016.

A. Guille and C. Favre. Mention-anomaly-based event detection and tracking in twitter. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pages 375–382. IEEE, 2014.

A. Guille, H. Hacid, C. Favre, and D. A. Zighed. Information diffusion in online social networks: A survey. *SIGMOD Rec.*, 42(2):17–28, July 2013. ISSN 0163-5808. doi: 10.1145/2503792.2503797. URL `https://doi.org/10.1145/2503792.2503797`.

H. Hamdan, F. Béchet, and P. Bellot. Experiments with dbpedia, wordnet and sentiwordnet as resources for sentiment analysis in micro-blogging. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 455–459, 2013.

Y. Han, S. Karunasekera, C. Leckie, and A. Harwood. Multi-spatial scale event detection from geo-tagged tweet streams via power-law verification. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1131–1136. IEEE, 2019.

Z. S. Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

M. Hasan, M. A. Orgun, and R. Schwitter. A survey on real-time event detection from the twitter data stream. *Journal of Information Science*, 44(4):443–463, 2018.

M. Hasan, M. A. Orgun, and R. Schwitter. Real-time event detection from the twitter data stream using the twitternews+ framework. *Information Processing & Management*, 56(3):1146–1165, 2019.

M. Heckenmüller, D. Narita, G. Klepper, et al. Global availability of phosphorus and its implications for global food supply: an economic overview. Technical report, Kiel working paper, 2014.

M. Henckens, E. Van Ierland, P. Driessen, and E. Worrell. Mineral resources: Geological scarcity, market price trends, and future generations. *Resources Policy*, 49: 102–111, 2016.

A. Hernández-Fuentes and A. Monnier. Twitter as a source of information? practices of journalists working for the french national press. *Journalism Practice*, pages 1–18, 2020.

H. Hettiarachchi, M. Adedoyin-Olowe, J. Bhogal, and M. M. Gaber. Embed2detect: Temporally clustered embedded words for event detection in social media. *Machine Learning*, pages 1–39, 2021.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997.

C. Honey and S. C. Herring. Beyond microblogging: Conversation and collaboration via twitter. In *2009 42nd Hawaii International Conference on System Sciences*, pages 1–10. Ieee, 2009.

Y. Hong, W. Zhou, J. Zhang, G. Zhou, and Q. Zhu. Self-regulation: Employing a generative adversarial network to improve event detection. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 515–526, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1048. URL `https://www.aclweb.org/anthology/P18-1048`.

M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020.

H. Hotelling. The economics of exhaustible resources. *Journal of political Economy*, 39(2):137–175, 1931.

M. Huchet-Bourdon. Agricultural commodity price volatility. (52), 2011. doi: https://doi.org/https://doi.org/10.1787/5kg0t00nrthc-en. URL `https://www.oecd-ilibrary.org/content/paper/5kg0t00nrthc-en`.

D. S. Jacks, K. H. O'rourke, and J. G. Williamson. Commodity price volatility and world market integration since 1700. *Review of Economics and Statistics*, 93(3): 800–813, 2011.

S. Jasinski. Phosphate rock statistical information. *USGS Minerals Yearbook*.

A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65, 2007.

X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu. Tiny{bert}: Distilling {bert} for natural language understanding, 2020. URL `https://openreview.net/forum?id=rJx0Q6EFPB`.

K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 1972.

P. Jürgens, A. Jungherr, and H. Schoen. Small worlds with a difference: New gatekeepers and the filtering of political information on twitter. 06 2011.

N. Kaldor. Speculation and economic stability. In *The economics of futures trading*, pages 111–123. Springer, 1976.

R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-thought vectors. *arXiv preprint arXiv:1506.06726*, 2015.

D. Kodelja, R. Besançon, and O. Ferret. *Exploiting a More Global Context for Event Detection Through Bootstrapping*, pages 763–770. 04 2019. ISBN 978-3-030-15711-1. doi: 10.1007/978-3-030-15712-8_51.

G. Kuang, Y. Guo, Y. Liu, L. Pang, Z. Yu, X. Cheng, J. Liu, and X. Yu. Bursty event detection via multichannel feature alignment. In *Proceedings of the 2020 5th International Conference on Big Data and Computing*, pages 39–45, 2020.

S. Kumar, H. Liu, S. Mehta, and L. V. Subramaniam. From tweets to events: exploring a scalable solution for twitter streams. *arXiv preprint arXiv:1405.1392*, 2014.

H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600, 2010.

W. C. Labys, A. Achouch, and M. Terraza. Metal prices and the business cycle. *Resources Policy*, 25(4):229–238, 1999.

J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977. ISSN 0006341X, 15410420. URL `http://www.jstor.org/stable/2529310`.

J. Leveling, M. Edelbrock, and B. Otto. Big data analytics for supply chain management. In *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 918–922. IEEE, 2014.

C. Li, A. Sun, and A. Datta. Twevent: Segment-based event detection from tweets. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, page 155–164, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450311564. doi: 10.1145/2396761.2396785. URL `https://doi.org/10.1145/2396761.2396785`.

Q. Li, A. Nourbakhsh, S. Shah, and X. Liu. Real-time novel event detection from social media. In *2017 IEEE 33Rd international conference on data engineering (ICDE)*, pages 1129–1139. IEEE, 2017.

Y. Liu and M. Lapata. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*, 2019.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

C. Maigrot, E. Kijak, and V. Claveau. Médias traditionnels, médias sociaux : caractériser la réinformation. In *TALN 2016 - 23ème Conférence sur le Traitement Automatique des Langues Naturelles*, Paris, France, July 2016. URL `https://hal.inria.fr/hal-01349871`.

C. Maigrot, V. Claveau, and E. Kijak. Fusion-based multimodal detection of hoaxes in social networks. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 222–229, 2018. doi: 10.1109/WI.2018.00-86.

E. Maître, Z. Chemli, M. Chevalier, B. Dousset, J.-P. Gitto, and O. Teste. Event detection and time series alignment to improve stock market forecasting. In *Joint Conference of the Information Retrieval Communities in Europe (CIRCLE 2020)*, volume 2621, pages 1–5. CEUR-WS. org, 2020.

E. Maître, Z. Chemli, M. Chevalier, B. Dousset, J. Gitto, and O. Teste. Étude de l'influence des représentations textuelles sur la détection d'évènements non supervisée dans des flux de données. In S. Nurcan, M. Savonnet, and T. Grison, editors, *Actes du XXXIXème Congrès INFORSID, Dijon, France, June 1-4, 2021*, pages 23–38, 2021. URL `http://inforsid.fr/actes/2021/INFORSID_2021_p23-38.pdf`.

E. Maitre, G. Ramalho Sena, Z. Chemli, M. Chevalier, B. Dousset, J.-P. Gitto, and O. Teste. The investigation of an event-based approach to improve commodities supply chain management. *Brazilian Journal of Operations Production Management*, 19(2):1–19, Apr. 2022. doi: 10.14488/BJOPM.2022.005. URL `https://bjopm.emnuvens.com.br/bjopm/article/view/1160`.

B. Mazoyer, J. Cage, C. Hudelot, and M.-L. Viaud. Real-time collection of reliable and representative tweets datasets related to news events. In *First International Workshop on Analysis of Broad Dynamic Topics over Social Media (BroDyn 2018) co-located with the 40th European Conference on Information Retrieval (ECIR 2018)*, Grenoble, France, Mar. 2018. URL `https://hal-centralesupelec.archives-ouvertes.fr/hal-02321957`.

B. Mazoyer, J. Cagé, N. Hervé, and C. Hudelot. A french corpus for event detection on twitter. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6220–6227, 2020a.

B. Mazoyer, N. Hervé, C. Hudelot, and J. Cage. Représentations lexicales pour la détection non supervisée d'événements dans un flux de tweets : étude sur des corpus français et anglais. In *Extraction et Gestion des connaissances, EGC 2020*, Jan. 2020b.

S. C. McGregor and L. Molyneux. Twitter's influence on news judgment: An experiment among journalists. *Journalism*, 21(5):597–613, 2020.

A. J. McMinn and J. M. Jose. Real-time entity-based event detection for twitter. In *International conference of the cross-language evaluation forum for european languages*, pages 65–77. Springer, 2015.

A. J. McMinn, Y. Moshfeghi, and J. M. Jose. Building a large-scale corpus for evaluating event detection on twitter. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 409–418, 2013.

Q. McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, June 1947. doi: 10.1007/BF02295996. URL `https://ideas.repec.org/a/spr/psycho/v12y1947i2p153-157.html`.

A. A. Mensah. Dynamique et comportements stratégiques sur le marché international du phosphate. *Mondes en développement*, (2):37–56, 2003.

M. C. Mew, G. Steiner, and B. Geissler. Phosphorus supply chain—scientific, technical, and economic foundations: a transdisciplinary orientation. *Sustainability*, 10(4): 1087, 2018.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

D. Miller. Leveraging BERT for extractive text summarization on lectures. *CoRR*, abs/1906.04165, 2019. URL `http://arxiv.org/abs/1906.04165`.

K. Morabia, N. L. Bhanu Murthy, A. Malapati, and S. Samant. SEDTWik: Segmentation-based event detection from tweets using Wikipedia. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 77–85, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-3011. URL `https://www.aclweb.org/anthology/N19-3011`.

M. Naaman, H. Becker, and L. Gravano. Hip and trendy: Characterizing emerging trends on twitter. *JASIST*, 62:902–918, 05 2011. doi: 10.1002/asi.21489.

D. Q. Nguyen, T. Vu, and A. T. Nguyen. Bertweet: A pre-trained language model for english tweets. *CoRR*, abs/2005.10200, 2020. URL `https://arxiv.org/abs/2005.10200`.

T. H. Nguyen and R. Grishman. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 365–371, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2060. URL `https://www.aclweb.org/anthology/P15-2060`.

N. Nicholas Taleb. The black swan: The impact of the highly improbable. *Victoria*, 250:595–7955, 2015.

D. Nolasco and J. Oliveira. Subevents detection through topic modeling in social media posts. *Future Generation Comp. Syst.*, 93:290–303, 2019.

P. Nora. L'évènement monstre. In *Communications*, pages 162–172, 1972.

N. Oliveira, P. Cortez, and N. Areal. The impact of microblogging data for stock market prediction: Using twitter to predict returns, volatility, trading volume and

survey sentiment indices. *Expert Systems with Applications*, 73, 12 2016. doi: 10.1016/j.eswa.2016.12.036.

R. B. Palazzi, A. C. F. Pinto, M. C. Klotzle, and E. M. De Oliveira. Can we still blame index funds for the price movements in the agricultural commodities market? *International Review of Economics & Finance*, 65:84–93, 2020.

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi: 10.1109/TKDE.2009.191.

A. Pandya, M. Oussalah, P. Kostakos, and U. Fatima. Mated: Metadata-assisted twitter event detection system. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 402–414. Springer, 2020.

S. Petrović, M. Osborne, and V. Lavrenko. Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, page 181–189, USA, 2010. Association for Computational Linguistics. ISBN 1932432655.

R. S. Pindyck, D. L. Rubinfeld, and E. Rabasco. *Microeconomia*. Pearson Italia, 2013.

M. M. A. Qudar and V. Mago. Tweetbert: A pretrained language representation model for twitter text analysis. *CoRR*, abs/2010.11091, 2020. URL `https://arxiv.org/abs/2010.11091`.

M. Quezada and B. Poblete. A lightweight representation of news events on social media. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 1049–1052, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361729. doi: 10.1145/3331184.3331300. URL `https://doi.org/10.1145/3331184.3331300`.

M. Radetzki and L. Wårell. *The Geography of Commodity Production and Trade*, page 28–55. Cambridge University Press, 2 edition, 2016. doi: 10.1017/9781316416945.003.

N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL `https://arxiv.org/abs/1908.10084`.

Ø. Repp and H. Ramampiaro. Extracting news events from microblogs. *Journal of Statistics and Management Systems*, 21(4):695–723, 2018.

A. N. Rezitis and M. Sassi. Commodity food prices: Review and empirics. *Economics Research International*, 2013, 2013.

Z. Saeed, R. Abbasi, M. Razzak, and G. Xu. Event detection in twitter stream using weighted dynamic heartbeat graph approach [application notes]. *IEEE Computational Intelligence Magazine*, 14:29–38, 08 2019a. doi: 10.1109/MCI.2019.2919395.

Z. Saeed, R. A. Abbasi, O. Maqbool, A. Sadaf, I. Razzak, A. Daud, N. R. Aljohani, and G. Xu. What's happening around the world? a survey and framework on event detection techniques on twitter. *Journal of Grid Computing*, 17(2):279–312, 2019b.

T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860, 2010.

V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

M. Shao, J. Li, F. Chen, H. Huang, S. Zhang, and X. Chen. An efficient approach to event detection and forecasting in dynamic multivariate social media networks. pages 1631–1639, 04 2017. doi: 10.1145/3038912.3052588.

R. J. Shiller. From efficient markets theory to behavioral finance. *Journal of economic perspectives*, 17(1):83–104, 2003.

R. Soni and S. Pal. Microblog retrieval for disaster relief: How to create ground truths? In *SMERP@ ECIR*, pages 42–51, 2017.

D. Spina, J. Gonzalo, and E. Amigó. Learning similarity functions for topic detection in online reputation monitoring. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 527–536, 2014.

R. Sprugnoli and S. Tonelli. One, no one and one hundred thousand events: Defining and processing events in an inter-disciplinary perspective. *Natural Language Engineering*, 23(4):485–506, 2017.

A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.

A. Swasy. A little birdie told me: Factors that influence the diffusion of twitter in newsrooms. *Journal of Broadcasting & Electronic Media*, 60(4):643–656, 2016.

S. J. Van Kauwenbergh. *World phosphate rock reserves and resources*. IFDC Muscle Shoals, 2010.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

G. von Nordheim, K. Boczek, and L. Koppers. Sourcing the sources. *Digital Journalism*, 6(7):807–828, 2018. doi: 10.1080/21670811.2018.1490658. URL `https://doi.org/10.1080/21670811.2018.1490658`.

O. Weber, J. Delince, Y. Duan, L. Maene, T. Mcdaniels, M. Mew, U. Schneidewind, and G. Steiner. *Trade and Finance as Cross-Cutting Issues in the Global Phosphate and Fertilizer Market*, pages 275–294. 03 2014. ISBN 978-94-007-7249-6. doi: 10.1007/978-94-007-7250-2_7.

F.-W. Wellmer and M. Dalheimer. The feedback control cycle as regulator of past and future mineral supply. *Mineralium Deposita*, 47(7):713–729, 2012.

J. Weng and B.-S. Lee. Event detection in twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5, 2011.

Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Y. Yang, T. Pierce, and J. Carbonell. A study of retrospective and on-line event detection. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 28–36, 1998.

A. Yeh. More accurate tests for the statistical significance of result differences. *arXiv preprint cs/0008005*, 2000.

X. Zhang, H. Fuehres, and P. A. Gloor. Predicting stock market indicators through twitter "i hope it is not as bad as i fear". *Procedia-Social and Behavioral Sciences*, 26:55–62, 2011.

D. Zhou, X. Zhang, and Y. He. Event extraction from twitter using non-parametric bayesian mixture model with word embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 808–817, 2017.

A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter. Detection and resolution of rumours in social media: A survey. 51(2), Feb. 2018. ISSN 0360-0300. doi: 10.1145/3161603. URL `https://doi.org/10.1145/3161603`.

Appendix A

Appendix

Impact side	Role	Sector	Player	Geographical zone
Context	Académie et Recherche	Agriculture	INRA	France
		Phosphore	Global Phosphorus Research Initiative	International
			Global TraPs	International
	Institution, Association	Bank	World Bank	International
		Fertilizers	IFA	International
		Bank	IFM	International
		Raw materials	ERMA - European Raw Materials Alliance	Europe
		Ressources research	USGS	USA
Demand	Industry	Agro business	Mars	USA
			Agropur Cooperative	Canada
			Ajinomoto	Japan
			Anheuser-Busch InBev	USA
			Archer Daniels Midland Company	USA
			Arla Foods	Europe
			Asahi Group	Japan
			Associated British Foods	UK
			Bacardi	USA
			Barry Callebaut	Switzerland
			Boparan Holdings	UK
			Brf Brasil Foods	Brazil
			Bunge	USA
			Campbell Soup Company	USA
			Cargill	USA
			Carlsberg	Dannemark
			China Mengniu Dairy Company	China

Figure A.1: Players identified in Section 6.4

CHS Inc.	USA
Coca-Cola Bottlers Japan	Japan
Coca-Cola European Partners	Europe
Coca-Cola HBC	Europe
ConAgra Brands	USA
Constellation Brands	USA
Dairy Farmers of America	USA
Danish Crown	Dannemark
Danone	Europe
Dean Foods Company	USA
Diageo	UK
DMK Deutsches Milchkontor	Europe
Dole Food Company, Inc.	USA
E & J Gallo Winery	USA
Femsa	USA
Ferrero	Europe
Flowers Foods	USA
Fonterra	New Zealand
General Mills Inc.	USA
Grupo Bimbo (Mexico)	USA
Hangzhou Wahaha Group	China
Heineken	Europe
Hormel Foods Corporation	USA
Ingredion Inc.	USA
Ito En	Japan
Itoham Yonekyu	Japan
J R Simplot	USA
Jacobs Douwe Egberts	Europe
JBS	Brazil
Kellogg Company	USA
Kerry Group	Europe

Figure A.2: Players identified in Section 6.4 - 2

Keurig Dr Pepper	USA
Kewpie Corporation	Japan
Kirin Holdings	Japan
Kraft Heinz Company	USA
Lactalis	Europe
Land O' Lakes Inc.	USA
Lindt & Sprungli	Switzerland
LVMH	Europe
Marfrig Group	Brazil
Maruha Nichiro Corporation	Japan
McCain Foods Ltd	Canada
McCormick Corporation	USA
Meiji Holdings	Japan
Molson Coors Brewing Company	USA
Mondelez International	USA
Morinaga Milk Industry	Japan
Muller Group	Europe
Nestle	Switzerland
NH Foods	Japan
Nisshin Seifun Group	Japan
Nissin Foods Group	Japan
Nissui	Japan
Oetker Group	Europe
Olam International	Singapoure
OSI Group	USA
Parmalat	Europe
PepsiCo Inc.	USA
Perdue Farms	USA
Pernod Ricard	Europe
Post Holdings	USA
Red Bull	Europe
Royal FrieslandCampina	Europe
Sapporo Holdings	Japan
Saputo	Canada

Figure A.3: Players identified in Section 6.4 - 3

			Savencia Fromage & Dairy	Europe
			Schreiber Foods	USA
			Smithfield Foods/WH Group	USA
			Sodiaal	Europe
			Sudzucker	Europe
			Suntory	Japan
			ThaiBev	Thailand
			The Coca-Cola Company	USA
			The Hershey Company	USA
			The JM Smucker Company	USA
			Total Produce	Europe
			Treehouse Foods	USA
			Tsingtao Brewery	China
			Tyson Foods	USA
			Unilever	Europe
			Vion	Europe
			Yamazaki Baking	Japan
			Yili Group	China
Supply	Industry	Mining and processing	Apatit	Kola (Russia)
			China Molybdenum Co.	Ltd Chapadão (Goiàs, Brazil)
			EuroChem	Kovdorskiy GOK (Russia)
			Foskor	Phalaborwa (South Africa)
			Mosaic Co.	Four Corners (Florida, USA)
				Hopewell (Florida, USA)
				South Ford Meade (Florida, USA)
				South Pasture (Florida, USA)

Figure A.4: Players identified in Section 6.4 - 4

				Wingate Creek (Florida, USA)
			Nutrien (merger of Agrium and Potash Corp.)	Dry Valley (Idaho, US)
				Swift Creek (Florida, US)
			OCP	Benguérir (Morocco)
				Boucraâ (Sahara)
				Khouribga (Morocco)
				Youssoufia (Morocco)
			P4 Production, LLC.	Blackfoot Bridge (Idaho, US)
			Sinochem Yunlong Co., Ltd.	Aurora (North Carolina, US)
			Vale	Bayóvar (Sechura, Peru)
				Catalão (Goiás, Brazil)
			Yara	Siilinjärvi (Finland)
		Phosphorus Production	5-Continent Phosphorus Co. Ltd.	China
			Changzhou Qishuyan Fine Chemical Co. Ltd	China
			Kazphosphate LLC	Kazakhstan
			Taj Pharmaceuticals Ltd.	India
			UPL Europe Ltd.	India
			Viet Hong Chemical and Trading Co. Ltd	Vietnam
			Yunphos (Taixing) Chemical Co., Ltd.	China

Figure A.5: Players identified in Section 6.4 - 5

Appendix B

Appendix

Sample tweets	‘China GDP economy grows by 4 8 in first quarter despite headwinds China Q1 GDP data Q1 GDP YoY 4 8 Est’ ‘LIVE China releases first quarter GDP data’ ‘Breaking Chinas economy accelerated 4 8 in the first quarter beatingestimates’ ‘China Quarterly GrossDomesticProduct QoQ Chinese gross domestic product GDP climbs more than expected in 1Q 2’ ‘Unbelievable China s Q1 GDP expands 4 8 y y better than f cast’
Named entities	[('China', 181), ('4 8', 49), ('4 8 year', 48), ('s first quarter', 34), ('4', 30), ('the first quarter of 2022', 29), ('first quarter', 28), ('the first quarter', 27), ('4 8 percent', 16), ('a year earlier', 14)]

Figure B.1: China published some economic results

Sample tweets	‘Shanghai China reports three dead in latest Covidoutbreak’ ‘China s Shanghai reports first COVID deaths since start of lockdown saralbharatnews COVID China chinacovid’ ‘Covid 19 first deaths in Shanghai since the start of containment asia pacific shanghai china’ ‘Chinas Shanghai Reports 3 Covid Deaths During RecentLockdown’ ‘China Posts Faster Growth That Masks Hit From Covid Lockdowns’
Named entities	[('Shanghai', 48), ('China', 48), ('three', 34), ('first', 18), ('Covid Lockdowns', 7), ('Covidoutbreak', 5), ('Covid', 5), ('3', 4), ('Covid Deaths', 4), ('Shanghai China', 4)]

Figure B.2: People are discussing the lockdown in Shangai and the new covid outbreak.

Sample tweets	‘Lets hope Elons first flight to Mars goes a little something like this’ ‘I mean it s one banana Michael what could it cost 10 dollars Elon Musk says almost anyone can afford 100 00 ‘ ‘Don t worry Elon Musk is planning to send Modizi to Mars’ ‘Elon Musk and Jeff Bezos are both planning trips to Mars However Senator Sanders of Vermont doesn t seem to be im’ ‘Elon Musk Says Almost Anyone Can Afford 100 000 a Hypothetical Price Point for a SpaceX Ticket to Mars’
Named entities	[('Mars', 10), ('Elon Musk', 6), ('one', 1), ('Michael', 1), ('10 dollars', 1), ('100', 1), ('Jeff Bezos', 1), ('Sanders', 1), ('Vermont', 1), ('100k', 1)]

Figure B.3: Some discussions about Elon Musk, SpaceX, and going to Mars.

Sample tweets	‘Terrebonne vs Bourgeois 2022 High School Baseball LIVE STREAM Watch Live Game The Ter’ ‘Paxton vs Crestview 2022 High School Baseball LIVE STREAM Watch Live Game The Paxton’ ‘Baldwin vs Parker 2022 High School Baseball LIVE STREAM Watch Live Game The Baldwin ‘ ‘Non Varsity Opponent vs Sayre 2022 High School Baseball LIVE STREAM Watch Live Game T’ ‘North Pocono vs Scranton 2022 High School Baseball LIVE STREAM Watch Live Game The No’
Named entities	[('2022', 64), ('Omaha', 3), ('Clinton', 3), ('Sharon Mutual', 2), ('Creighton', 2), ('Smyrna Beach', 2), ('University 2022 High School Baseball LIVE STREAM Watch Live Game', 2), ('Sheboygan South', 2), ('Green Bay West', 2), ('Douglas County', 2)]

Figure B.4: Multiple event clusters are about high school baseball games. We are unsure which of the provided keyword lead to this.

Table B.1: The keywords we used during the experiment with Twitter's stream. These results are derived from the tables presented in appendix A.

morocco	china	phosphate
phosphorus	fertilizer	inra
global phosphorus	global taps	world bank
ifa	ifm	erma
european raw materials alliance	usgs	mars
agropur cooperative	ajinomoto	anheuser-busch inbev
archer daniels midland company	arla foods	asahi group
associated british foods	bacardi	barry callebaut
boparan holdings	brf brasil foods	bunge
campbell soup company	cargill	carlsberg
china mengniu dairy company	chs inc.	coca-cola bottlers japan
coca-cola european partners	coca-cola hbc	conagra brands
constellation brands	dairy farmers of america	danish crown
danone	dean foods company	diageo
dmk deutsches milchkontor	dole food company	E & J gallo winery
femsa	ferrero	flowers food
fonterra	general mills inc.	grupo bimbo
hangzhou wahaha group	heineken	hormel foods corporation
yunphos	j r simplot	jacobs douwe egberts
jbs	kellogg company	kerry group
keurig dr pepper	kewpie corporation	kirin holdings
kraft heinz company	lactalis	lindt
lvmh	marfrig group	maruha nichiro corporation
mccain foods ltd	mccormick coporation	meiji holdings
molson coors brewing company	mondelez international	morinaga milk industry
muller group	nestle	nh foods
nisshin seifun group	nissui	oetker group
olam international	osi group	parmalat
pepsico inc.	perdue farms	pernod ricard
post holdings	red bull	roayl frieslandcampina
sapporo holdings	saputo	savencia fromage & diary
schreiber foods	smithfield foods/wh group	sodiaal
sudzucker	suntory	thaibev
the coca-cola company	coca-cola company	hershey company
jm smucker company	total energy	treehouse foods
tsingtao brewery	tyson foods	unilever
vion	yamazaki baking	yili goup
apatit	china molybdenum co.	eurochem
foskor	mosaic co.	nutrien
ocp	P4 production	sinochem yunlong co.
vale	yara	5-continent
phosphorus co. ltd.	changzhou oishuyan	kazphosphate
taj pharmaceuticals	upl europe	viet hong chemical
ingredion inc	ito enitoham yonekyun	

Résumé en français

L'objectif de cette thèse est de mettre en place un système de détection d'évènements sur les réseaux sociaux permettant d'assister les personnes en charge de prises de décisions dans des contextes industriels. Le but est de créer un système de détection d'évènement permettant de détecter des évènements à la fois ciblés, propres à des domaines particuliers, mais aussi des évènements généraux. En particulier, nous nous intéressons à l'application de ce système aux chaînes d'approvisionnements et plus particulièrement celles liées aux matières premières. Le défi est de mettre en place un tel système de détection, mais aussi de déterminer quels sont les évènements potentiellement impactant dans ces contextes. Cette synthèse résume les différentes étapes des recherches menées pour répondre à ces problématiques.

Architecture d'un système de détection d'évènements

Dans un premier temps, nous introduisons les différents éléments nécessaires à la constitution d'un système de détection d'évènements. Ces systèmes sont classiquement constitués d'une étape de filtrage et de nettoyage des données, permettant de s'assurer de la qualité des données traitées par le reste du système. Ensuite, ces données sont représentées de manière à pouvoir être regroupées par similarité. Une fois ces regroupements de données établis, ils sont analysés de manière à savoir si les documents les constituants traitent d'un évènement ou non. Finalement, l'évolution dans le temps de ces évènements est suivie. Nous avons proposé au cours de cette thèse d'étudier les problématiques propres à chacune de ces étapes.

Représentations textuelles de documents issus des réseaux sociaux

Nous avons comparé différentes méthodes de représentations des données textuelles, dans le contexte de notre système de détection d'évènements. Nous avons comparé les performances de notre système de détection à l'algorithme First Story Detection (FSD), un algorithme ayant les mêmes objectifs. Nous avons d'abord conclu que le système que nous proposons est plus performant que le FSD, mais aussi que les

architectures récentes de réseaux de neurones sont plus performantes que TF-IDF dans notre contexte, contrairement à ce qui avait été montré dans le contexte du FSD. Nous avons ensuite proposé de combiner différentes représentations textuelles afin d'exploiter conjointement leurs forces.

Détection d'évènement, suivi et évaluation

Nous avons proposé des approches pour les composantes d'analyse de regroupement de documents ainsi que pour le suivi de l'évolution de ces évènements. En particulier, nous utilisons l'entropie et la diversité d'utilisateurs introduits dans [Rajouter les citations] pour évaluer les regroupements. Nous suivons ensuite leur évolution au cours du temps en faisant des comparaisons entre regroupements à des instants différents, afin de créer des chaînes de regroupements. Enfin, nous avons étudier comment évaluer des systèmes de détection d'évènements dans des contextes où seulement peu de données annotées par des humains sont disponibles. Nous avons proposé une méthode permettant d'évaluer automatiquement les systèmes de détection d'évènement en exploitant des données partiellement annotées.

Application au contexte des matières premières

Afin de spécifier les types d'évènements à superviser, nous avons mené une étude historique des évènements ayant impacté le cours des matières premières. En particulier, nous nous sommes focalisés sur le phosphate, une matière première stratégique. Nous avons étudié les différents facteurs ayant une influence, proposé une méthode reproductible pouvant être appliquée à d'autres matières premières ou à d'autres domaines. Enfin, nous avons dressé une liste d'éléments à superviser pour permettre aux experts d'anticiper les variations des cours.