Étude de l'influence des représentations textuelles sur la détection d'évènements dans des flux de données

A study of the influence of textual representation on event detection in data streams

Elliot MAÎTRE[12], Max CHEVALIER[1], Bernard DOUSSET[1], Jean-Philippe GITTO[2], Olivier TESTE[1]

[1] IRIT, 118, Route de Narbonne, 31062 cedex 04, Toulouse, France, prenom.nom@irit.com
[2] Scalian, 22, bd Déodat de Séverac, 31770 Colomiers, France, prenom.nom@scalian.com

RÉSUMÉ. La détection d'évènements à partir des données postées sur internet est un sujet important de la recherche d'information. Les sources de données potentiellement intéressantes sont multiples et peuvent prendre la forme de flux de données textuelles plus ou moins structurées. Nous étudions dans cet article la détection d'évènements dans les flux de données textuelles et plus particulièrement l'impact de la représentation du texte sur la qualité des évènements détectés. Nous comparons différentes approches de traitement du langage dans deux contextes: supervisé et non supervisé. Nous étudions la question de l'efficacité des modèles basés sur les architectures Transformer pour la détection d'évènements dans les documents courts. Cette étude nous permet de conclure que, contrairement à ce qui avait pu être précédemment montré, les architectures Transformer peuvent être compétitives par rapport aux méthodes classiques.

ABSTRACT. Detection of real-world events using online data sources is a trending topic in the information retrieval domain. Multiple data sources are potentially of interest and some of them are data streams. There are multiple data sources that are potentially interesting, and some of them are textual data streams, structured or unstructured. We propose to analyse the problem of event detection from text data stream and to focus particularly on the importance of the representation of the textual data. To do so, we compare multiple approaches in different contexts: supervised and unsupervised. We focus on the performances of Transformer-based architectures for event detection on short text documents, and we conclude that, contrary to previous studies, these architectures can be competitive compared to classical methods.

MOTS-CLÉS. Recherche d'Information, Détection d'évènements, Traitement automatique du langage naturel, Partitionnement de données.

KEYWORDS. Information Retrieval, Event Detection, Natural Language Processing, Clustering.

Introduction

De nombreux évènements se produisent constamment et sont à l'origine de perturbations potentiellement importantes dans de nombreux domaines. Si l'exemple de la pandémie liée au virus Sars-cov2 est évidemment l'évènement venant le plus naturellement à l'esprit cette année, d'autres types d'évènements peuvent aussi avoir des impacts importants, comme les évènements politiques (élections présidentielles aux États-Unis), les évènements météorologiques ou encore les catastrophes naturelles. Malgré la facilité actuelle d'accès aux masses d'information, il est difficile d'avoir une vue exhaustive de l'ensemble des évènements se déroulant sur la planète, notamment du fait de la profusion d'informations. De manière à faciliter cette tâche de veille, des systèmes automatisés ont vu le jour afin de détecter les contenus importants. Une des manières d'aborder cette tâche est la détection d'évènements dans les données textuelles [ATE 15], [HAS 18]. En effet, un des principaux vecteurs de communication de la presse et sur Internet de manière générale est les données textuelles. Il est possible d'accéder à ces documents textuels via des flux, qu'ils soient issus de réseaux sociaux ou de journaux. La difficulté qui se présente est alors de réussir à trouver les sources intéressantes, mais aussi d'être capable d'extraire l'information pertinente de ces flux.

Nous proposons au cours de cette étude une méthode de détection d'évènements dans les flux de données textuelles. Ce problème est très étudié dans la littérature [SAK 10], [WEN 11], [HAS 19] et

constitue un problème important de la fouille de données textuelles. Cette tâche peut se décomposer en différentes parties, notamment la détection, le suivi et l'extraction des évènements [ALL 12]. Différentes approches sont possibles pour chacune de ces tâches. Nous nous focalisons ici sur la tâche de détection d'évènements. Cette tâche est souvent abordée comme un problème de partitionnement, dynamique ou non, où chacun des partitionnements correspond à un évènement ou à une sous-partie d'un évènement [ALL 12].

Nous souhaitons évaluer la pertinence de l'utilisation des modèles de langage basés sur les Transformers, permettant d'exploiter le contenu sémantique des documents, qui ont prouvé leur efficacité dans de nombreux domaines du TALN (Traitement Automatisé du Langage Naturel) et qui tendent à remplacer les réseaux de neurones récurrents dans ces domaines, [CER 18], [REI 19] pour la détection d'évènements dans des textes courts. En effet, les performances de ces modèles n'ont pas été évaluées dans un cadre de partitionnement classique et ont même été évaluées comme moins performantes que TF-IDF (term frequency-inverse document frequency) dans le cadre d'un partitionnement dynamique [MAZ 20]. Nous cherchons à montrer l'intérêt du partitionnement classique par rapport au partitionnement dynamique dans ce contexte.

Afin de répondre à cette problématique, nous proposons une méthode de détection d'évènements dans des flux de données textuelles basée sur le partitionnement de données où le flux de données est découpé en fenêtres, de manière à pouvoir appliquer des algorithmes de partitionnement classiques, et ainsi s'extraire des contraintes imposées par le partitionnement dynamique. Cela permet de considérer l'ensemble des documents publiés au moment du partitionnement, et non de devoir travailler avec des informations fragmentaires au fil de l'eau. Cela présente un intérêt particulier pour les flux de textes courts, où l'information fragmentaire rend difficile l'identification des évènements. En considérant un groupe de documents, il est plus aisément possible de reconstituer le contenu de l'évènement. Il suffit ensuite de calibrer la taille des fenêtres en fonction de la granularité souhaitée. Ainsi, nous évaluons dans un premier temps l'impact du type (e.g. heures, nombre de documents) et de leur taille sur notre modèle. Nous comparons ensuite notre approche à des approches de partitionnement dynamiques reconnues afin d'en valider la pertinence. Enfin, nous comparons différentes méthodes de représentation des données textuelles. Plus particulièrement, nous nous intéressons aux approches basées sur les architectures Transformers qui sont actuellement reconnues comme disruptives dans le domaine du TALN mais n'ont pas encore prouvé leur efficacité pour des documents courts et peu structurés comme ceux issus des réseaux sociaux. Ces approches sont pourtant particulièrement intéressantes dans ce contexte, notamment par leur capacité à traiter les données hors du vocabulaire automatiquement, contrairement à aux approches comme TF-IDF ou Word2vec [MIK 13]. Cela peut constituer un atout majeur dans un contexte comme les réseaux sociaux où le vocabulaire évolue rapidement. Notre approche montre que, contrairement aux études précédemment menées [MAZ 20], les approches basées sur des architectures Transformers peuvent avoir des performances similaires aux approches classiques comme TF-IDF dans ce contexte. Cet article, complète l'article [MAI 21] en développant l'approche, les expériences et les résultats avec plus de détails. De plus, une expérience complémentaire concernant l'impact du type et de la taille des fenêtres est menée. Les résultats des expériences suivantes viennent par la suite compléter les résultats présentés dans l'article précédent, à la lumière des résultats obtenus au cours de cette expérience complémentaire.

Le reste de cet article s'organise de la manière suivante. La section 1 présente les travaux de la littérature. Ensuite, la section 2 détaille notre approche. Enfin, nous présentons et discutons nos résultats dans la section 3.

1. État de l’art

Dans un premier temps, nous présentons les différentes manières de représenter le contenu textuel, notamment les approches vectorielles. Dans un second temps, nous étudions différentes approches existantes pour la détection d'évènements à partir de textes, avec une attention particulière sur les documents issus des réseaux sociaux. Les réseaux sociaux ont la particularité d'être très réactifs aux

évènements, notamment Twitter. En effet, la structure de Twitter en fait un média similaire à un média d'actualité [KWA 10] et Twitter est un environnement idéal pour la dissémination d'informations [CAS 11]. Ce réseau social est utilisé à la fois par des particuliers que par des profesionnels tels que les journalistes [VON 18]. De ce fait, il est particulièrement intéressant comme média regroupant des informations issues de différentes sources, et peut être utilisé pour détecter des évènements ayant un intérêt pour de nombreux domaines, comme par exemple la bourse [MAI 20].

1.1. *Représentation du contenu textuel*

Les méthodes de représentation du contenu textuel constituent un des enjeux majeurs des travaux relatifs à la recherche d'informations [BAE 99]. La méthode constituant actuellement la référence est TF-IDF [JON 72] qui permet de prendre en compte l'importance des mots dans la représentation du document en pondérant chaque mot de manière inversement proportionnelle au nombre de documents dans lesquels il apparaît. Ainsi, un mot apparaissant dans un document alors qu'il n'apparaît que peu dans le corpus est considéré comme porteur de beaucoup d'informations. Sa pondération dans le cadre de TF-IDF est donc forte. Cette représentation est très utilisée, même de nos jours, dans la recherche d'information et obtient de très bonnes performances, même sur les textes courts du type réseaux sociaux.

Ces représentations statistiques sont actuellement complétées par des représentations vectorielles, appelées plongement de mots, basées sur des approches d'apprentissage profond. [MIK 13] introduisent le modèle Word2vec qui correspond à une approche neuronale permettant d'associer à un mot un vecteur, qui est calculé grâce au contexte dans lequel le mot apparaît dans le jeu d'entraînement. Ainsi, le vecteur représentant un mot contient de l'information à propos de celui-ci. L'hypothèse faite pour la constitution de ces vecteurs est que des mots dont l'utilisation contextuelle est proche, seront porteurs d'un sens similaire et donc seront représentés par un vecteur proche. Des variations existent, comme le modèle FastText [BOJ 16] qui découpe les mots en sous-mots, permettant de prendre en compte la construction des mots, notamment les suffixes et les préfixes. Les modèles les plus récents sont basés sur des architectures Transformers [VAS 17]. Le plus notable d'entre eux est BERT [DEV 18]. L'architecture de BERT peut s'appliquer à toutes les tâches grâce à une approche d'apprentissage par transfert (transfer learning) [PAN 10]. En effet, le modèle est d'abord pré-entraîné sur deux types de tâches, prédire les mots masqués dans une phrase et prédire la phrase suivante. Un affinage (fine-tuning) est ensuite possible sur la tâche spécifique pour laquelle le modèle doit être utilisé.

Tous ces modèles permettent de représenter des mots mais ne permettent pas nécessairement de représenter des phrases. Une des premières approches est Skip-Thought, proposée par [KIR 15]. C'est une architecture encodeur-décodeur, entraînée de manière non supervisée à prédire les phrases voisines d'une phrase donnée dans un texte. Une autre approche classique est l'utilisation de réseaux siamois, c'est-à-dire deux réseaux de neurones en parallèle, possédant la même architecture et les mêmes poids, mais qui ne prendront pas la même entrée [BRO 94]. C'est notamment ce qui a été proposé par [CON 17] avec leur modèle InferSent. C'est un réseau LSTM (Long short-term memory) bi-directionnel siamois entraîné de manière supervisée sur le jeu de données SNLI (The Stanford Natural Language Inference corpus) [BOW 15]. Ce jeu de données contient 570 000 paires de phrases annotées selon trois catégories : implication entre la première et la deuxième phrase, contradiction de la première avec la deuxième phrase, les phrases sont neutres entre elles. Un autre moyen de représenter les phrases est d'utiliser une architecture basée sur les Transformers [CER 18]. Universal Sentence Encoder (USE) est entraîné sur deux types de tâches, une supervisée, basée sur le jeu de données SNLI de la même manière qu'Infersent, et sur des tâches non supervisées, comme Skip-Thought. Les architectures Transformers peuvent aussi être utilisées sous forme de réseaux siamois. C'est notamment l'approche suivie dans Sentence BERT (S-BERT) présentée par [REI 19]. Cette approche consiste à créer un réseau siamois de deux modèles BERT qui seront entraînés avec l'objectif de produire des vecteurs similaires pour des phrases dont le sens est proche et des vecteurs dissimilaires pour des

phrases dont le sens est éloigné. Ensuite, une dernière couche de neurones est rajoutée, de manière à pouvoir être affinée sur des tâches spécifiques.

Dans la suite de ce papier, nous menons une étude comparative des modèles basés sur TF-IDF et ceux basés sur des architectures Transformers, en particulier S-BERT et USE. En effet, TF-IDF est la méthode historiquement la plus utilisée, et obtient de très bons résultats. Concernant les approches Transformers ont récemment obtenu des résultats supérieurs aux autres approches neuronales comme Word2Vec [MIK 13] dans le domaine du TALN. De plus, contrairement à ces approches, les modèles Transformers sont capables de gérer les mots hors du vocabulaire automatiquement, sans entraînement spécifique sur le corpus considéré, ce qui est particulièrement pertinent pour les réseaux sociaux, où le vocabulaire utilisé est très variable et évolue rapidement. Nous choisissons donc de nous orienter vers ces approches dans cette étude.

1.2. *La détection d'évènements*

La détection d'évènements sur les réseaux sociaux est une tâche de fouille de texte classique de la littérature [ALL 17]. Les réseaux sociaux sont particulièrement étudiés pour la détection d'évènements car ils sont très réactifs et des informations traitant du court terme ou du long terme y sont discutées [ZUB 18]. Le réseau le plus classiquement étudié est Twitter, car il est le plus performant pour la détection d'évènements [HAS 18].

La détection d'évènements est un dérivé de la détection et du suivi de sujet (TDT : Topic Detection and Tracking), et peut être divisée en différentes sous-tâches selon [ALL 12] : la segmentation de sujets, la détection de nouveaux sujets (FSD : First Story Detection), le partitionnement (Cluster Detection), le suivi et la détection de liens. Nous nous intéresserons plus particulièrement aux tâches de détection de nouveaux sujets et au partitionnement. Ces sous-tâches peuvent être abordées de différentes manières, se divisant en deux grandes catégories : document-pivot et feature-pivot. La première consiste à travailler à l'échelle du document tandis que la seconde travaille à l'échelle du mot ou de groupement de mots. Nous choisissons de nous focaliser sur les approches document-pivot. En effet, ces approches permettent de considérer l'ensemble du contenu textuel du document et d'exploiter un maximum de sens.

L'algorithme de FSD a d'abord été introduit par [ALL 00] dans le système Umass puis a été amélioré par [PET 10] introduisant l'algorithme de FSD avec LSH (Locality Sensitive Hashing), permettant d'accélérer la recherche de plus proches voisins. L'objectif de cette méthode est de détecter le premier document faisant référence à un évènement. Le problème est ici abordé comme un problème de clustering dynamique des nouveaux documents. Dans [REP 18], les auteurs proposent d'accélérer l'algorithme du FSD en utilisant une approche basée sur les "mini-batchs". Cette version de l'algorithme est présentée dans l'algorithme 1. [HAS 19] proposent d'utiliser l'algorithme de FSD pour évaluer la nouveauté d'un tweet et assignent ensuite le tweet à un cluster à l'aide de la différence entre ce tweet et la moyenne de représentation des clusters. Les représentations des tweets sont calculées à l'aide de TF-IDF. [MAZ 20] proposent d'utiliser la version de l'algorithme présentée dans l'algorithme 1 et de comparer les performances de Word2vec, TF-IDF, BERT et USE pour la détection de nouveaux sujets.

Algorithm 1: First Story Detection, [REP18]

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

Les auteurs de [BEC 11] proposent de grouper les tweets dans des clusters de messages similaires afin de déterminer quels messages parlent d'évènements ou non. Ils utilisent TF-IDF pour représenter les tweets puis calculent une similarité pour créer des clusters et les classer à l'aide d'un classifieur SVM (Machine à Vecteurs de Support). Dans [BOO 16], les auteurs prolongent les travaux de Becker et al. en utilisant un algorithme de clustering incrémental et en exploitant la sémantique des hashtags pour améliorer le clustering. Ils filtrent ensuite les évènements triviaux. [MCM 15] utilisent aussi TF-IDF pour représenter les tweets et appliquent ensuite un algorithme de clustering incrémental se basant sur des critères de similarité et de taille des tweets pour les regrouper. Ils couplent cela avec des méthodes de filtrage pour permettre le passage à l'échelle de l'algorithme.

Dans la suite de cet article, nous proposons de comparer les méthodes basées sur les architectures Transformers à la méthode TF-IDF très majoritairement utilisée dans la littérature de manière à en évaluer les performances dans un contexte de partitionnement classique. Nous comparons aussi les performances entre les contextes de partitionnement dynamique et de partitionnement classique. Pour cela, nous menons un comparatif similaire à celui proposé par [MAZ 20] avec notre méthode et nous comparons les résultats qu'ils obtiennent à ceux obtenus avec notre méthode.

2. Le moteur de détection d'évènements : EDF

Dans cette section, nous présentons tout d'abord l'approche proposée, puis en donnons une description formelle avant de présenter les différents algorithmes mis en œuvre au cours de notre approche, et enfin nous présentons le jeu de données utilisé par la suite.

2.1. *Description de l'approche*

Nous proposons de traiter le problème de la détection d'événements dans un flux de données textuelles comme une tâche de partitionnement. Cela nous permet de nous affranchir de la contrainte imposée par le partitionnement dynamique, c'est-à-dire que nous pouvons ainsi considérer tous les documents publiés au moment du partitionnement, et ne pas avoir à travailler avec des informations fragmentaires sur le flux de documents. Nous avons conçu la méthode pour qu'elle soit flexible, de sorte que n'importe quel modèle de représentation vectorielle du texte et n'importe quel algorithme de partitionnement classique peuvent être utilisés. Cette flexibilité est particulièrement intéressante car il est important de pouvoir adapter le couple modèle de représentation/algorithme de partitionnement, pour s'adapter à l'état de l'art qui évolue rapidement dans ces domaines. Pour être dans un contexte classique de partitionnement, nous divisons le flux de données en utilisant des fenêtres, c'est-à-dire des fenêtres de taille fixe (nombre fixe de documents) ou des fenêtres de temps fixe (documents publiés pendant une période de temps fixe, par exemple 1 heure). Cette approche permet de s'assurer que les documents regroupés ont une date de publication similaire, ce qui augmente les chances que les documents traitent réellement du même événement.

Nous nous intéressons à l'évaluation des performances de différents couples modèles de représentation/algorithmes de clustering. Pour ce faire, nous faisons les hypothèses suivantes : (1) tous les documents sont liés à un événement, (2) chaque document est associé à exactement un événement, (3), il y a un nombre inconnu d'évènements. Sous ces hypothèses, nous pouvons évaluer correctement les performances de chaque modèle de représentation. Ceci est courant dans la littérature [BEC 10], [BOO 16], [MAZ 20]. Aucun filtrage ne sera effectué sur les documents car ils sont tous liés à des événements. Dans une configuration plus proche de la réalité, des étapes de filtrage sont appliquées pour filtrer le spam et les documents inintéressants. Ici, nous effectuerons uniquement un nettoyage des données. Après l'étape de "partitionnement de documents", les regroupements sont généralement évalués pour déterminer s'ils traitent d'un événement ou d'une simple conversation, puis sont résumés pour être présentés à des humains. Ces étapes sont indépendantes de la phase de partitionnement et sont donc hors du cadre de ce papier. En tenant compte de ces hypothèses, nous présentons le processus de traitement à la Figure 1 et nous détaillerons de manière plus formelle chaque étape du processus dans la section suivante.

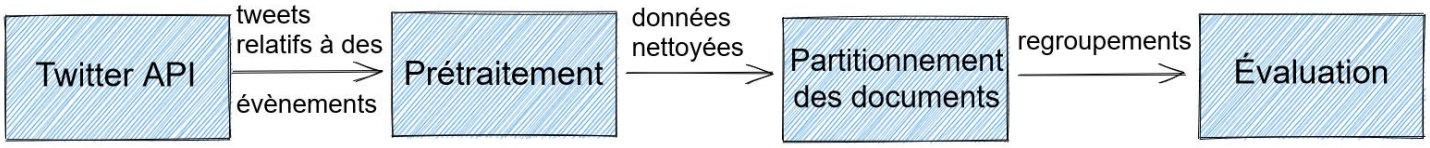

Figure 1. *Le processus de détection d'évènements mis en oeuvre dans EDF.*

2.2. *Description formelle du processus de regroupement*

En entrée, nous recevons un flux de documents relatifs à des évènements annotés $D = \{d_1, ..., d_N\}$. Nous définissons un document de la façon suivante : $\forall i \in [1..N]$, $d_i = (txt_i, dte_i, tag_i, url_i, src_i)$ où txt_i représente le contenu textuel du document, dte_i à la date de publication, tag_i aux tags and url_i aux urls partagées and src_i à la source qui poste le $i^{ème}$ document. Nous appliquons différentes étapes de nettoyage des données, décrites dans la suite du document. Ensuite, nous discrétisons le flux en utilisant des fenêtres fixes en termes de temps, ou de nombres de tweets, ce qui est habituel dans la littérature [MCM 15], [NAA 11], [GUI 14]. Cela permet de s'assurer que les documents regroupés ensemble ont une date de publication similaire, puisque des documents traitant d'un même évènement ont probablement été postés durant une période de temps similaire. Nous comparerons par la suite l'impact de la taille des fenêtres. Elles sont annotées $W = \{W^1, ..., W^m\}$ où $\forall k \in [1..m]$, $W^k = \{ d_1^k , ..., d_\tau^k \}$, où k fait référence à la $k^{ème}$ fenêtre, et τ au nombre de documents dans chaque fenêtre. Les fenêtres sont considérées indépendantes les unes des autres; i.e., $\forall k \in [1..m]$, $\forall l \in [1..m]$, $l \neq k$, $W^k \cap W^l = \emptyset$. Chaque fenêtre est partitionnée en groupes de documents similaires,

appelés regroupements notés *C*. Les documents dans W^k sont ainsi regroupés selon des métriques de similarité (par exemple basées sur le texte) pour obtenir un ensemble de regroupement tels que $\forall i \in [1..n]$, $\forall j \in [1..n]$, i $\neq$ j, $C_i^k \cap C_j^k = \emptyset$ et $\bigcup_{j=1}^{n} C_j^k = W^k$. Ainsi, notre méthode de détection d'événements est une succession de processus de partitionnement résultant de la discrétisation du flux à l'aide de fenêtres fixes et disjointes. Ceci diffère de l'algorithme FSD qui traite le problème de la détection d'événements comme un problème de partitionnement dynamique. Nous allons maintenant présenter les différents algorithmes et modèles utilisés pour chaque étape. Une description plus visuelle du processus pour une fenêtre est proposée dans la Figure 1 et un pseudo-algorithme est fourni à l'Algorithme 2.

Algorithm 2: EDF

input : threshold t, Window W,Representation model RM, Clustering Algorithm ClusteringAlg
output: ListClus, a list of Cluster for window W

$Repres \leftarrow [] ; SimMat \leftarrow [] ; ListReg \leftarrow [];$
foreach *document d in W* **do**
  $Repres(d) \leftarrow RepresModel(d);$
end
for (d_1, d_2) *in W* **do**
  $SimMat(d_1, d_2) \leftarrow Cosine(Repres(d_1), Repres(d_2))$
end
$ListClus \leftarrow ClusteringAlg(SimMat, t);$

2.3. *Algorithmes mis en oeuvre*

Nous proposons de comparer différents modèles de représentation de texte dans deux contextes différents : FSD, l'algorithme le plus classiquement utilisé dans la littérature, et EDF. Chacun de ces deux algorithmes est constitué de trois étapes majeures, à savoir la représentation du texte, le calcul de similarité entre les documents et le partitionnement des documents.

2.3.1. *Modèles de représentation du texte*

Nous comparons deux types de représentations de documents textuels : les approches statistiques, également appelées approches lexicales, et les modèles de langue basés sur des Transformers, également appelés approches sémantiques.

Approches lexicales - Nous utilisons TF-IDF, qui est le modèle de représentation du texte le plus courant en recherche d'information [BAE 99]. L'importance d'un mot dans un document est basée sur sa fréquence dans le document, mais aussi sa fréquence dans l'ensemble du corpus, selon l'hypothèse qu'un mot fréquent dans un document, mais pas dans le corpus est représentatif du document.

Approches sémantiques - Les approches de représentation sémantiques ont actuellement les meilleures performances dans le TALN, particulièrement celles basées sur les Transformers [VAS 17]. En particulier, nous comparons deux modèles de langue : S-BERT [REI 19] et Universal Sentence Encoder (USE) [CER 18]. Le principe de ces approches est d'obtenir des vecteurs de représentation similaires pour des phrases ayant un sens proche, et des vecteurs dissimilaires lorsque les phrases ont un sens éloigné.

2.3.2. *Partitionnement des données*

Pour chaque paire de documents et pour chaque modèle de représentation, nous calculons sa similarité pour constituer une matrice de similarité $S_{model,Wk}$ utilisée pour calculer les partitionnements. Nous avons choisi la similarité Cosinus car il s'agit de la mesure de similarité la plus courante en TALN [AGG 12]. Il est important de noter que les performances du partitionnement sont directement

affectées par les mesures de similarité, ce qui en fait une étape critique du processus de détection d'événements.

En utilisant ces similarités, les regroupements sont calculés en utilisant l'algorithme de Louvain [BLO 08], un algorithme de détection de communauté qui calcule automatiquement le nombre optimal de regroupements. Cet aspect est particulièrement important dans notre contexte de détection d'événements dans un domaine ouvert, dans lequel le nombre d'événements n'est pas connu à l'avance. Le seul paramètre dont cet algorithme a besoin est un seuil de similarité, qui sera différent pour chaque modèle de représentation.

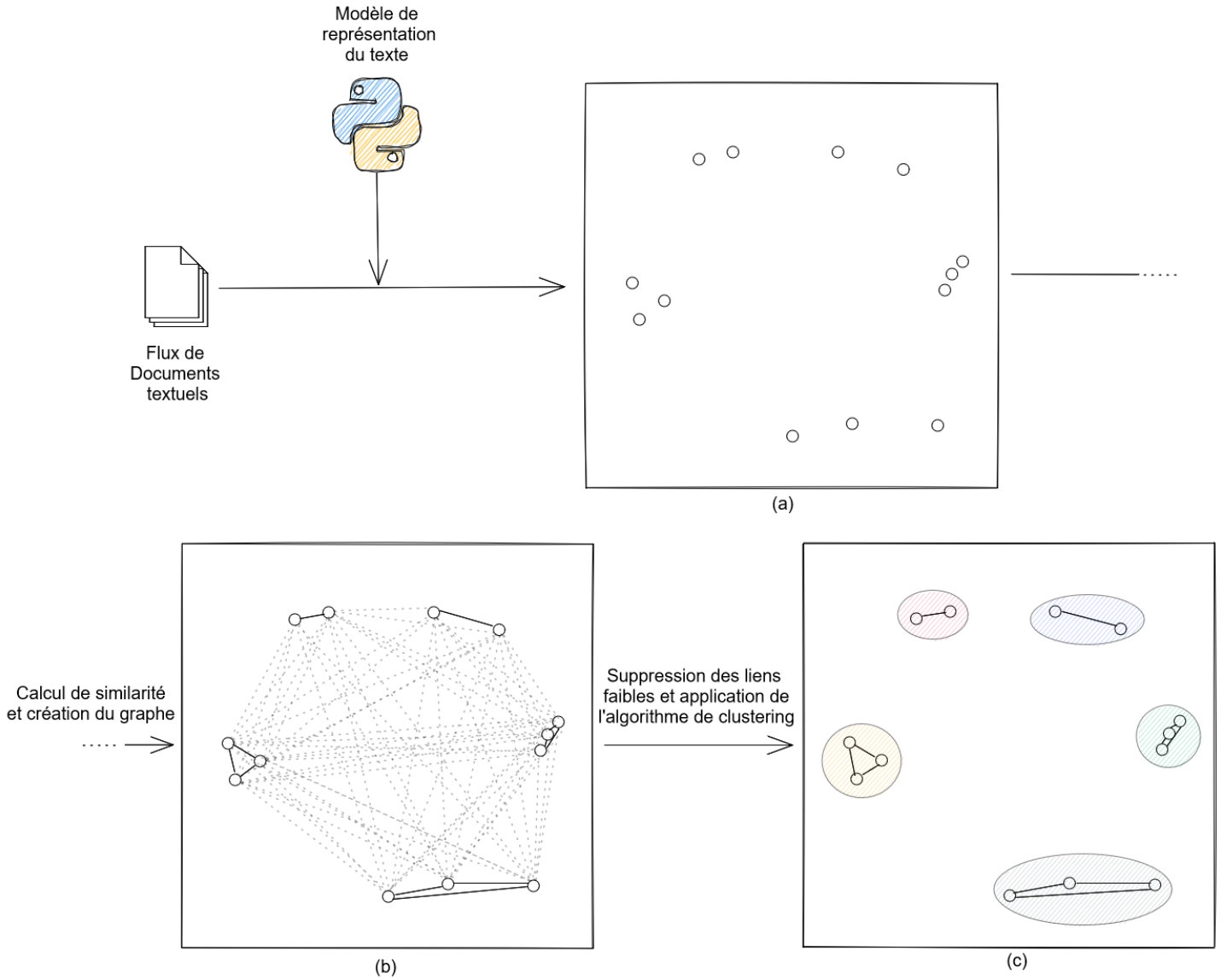

Figure 2. *Processus de traitement des données réalisé par EDF à l'échelle d'une fenêtre. (a) Représentation des documents dans l'espace. Chacun point est un document. (b) Création du graphe à partir de la matrice de similarité. Chaque document est un sommet, chaque arête est pondérée par la similarité entre documents. (c) Création des partitionnements, en supprimant les arêtes dont le poids est trop faible.*

Maintenant que nous avons présenté les différents algorithmes que nous utilisons, nous présentons le jeu de données sur lequel nous avons mené nos expériences.

2.4. *Présentation du jeu de données*

Nous utilisons Event2012 [MCM 13], un corpus de 120 millions de tweets, collectés entre le 10 octobre et le 7 novembre 2012 en utilisant l'API de streaming de Twitter. 159 952 tweets sont labellisés comme relatifs à des évènements et sont distribués dans 506 évènements, eux-mêmes répartis en 8 catégories. Nous travaillons uniquement sur la partie annotée du dataset afin de pouvoir évaluer les

modèles,nous considérons donc chaque regroupement de documents comme un évènement dans ce contexte. Afin de respecter les conditions d'utilisation de Twitter, seuls les identifiants des tweets sont partagés, permettant par la suite la récupération du contenu. Du fait de la suppression des tweets au cours du temps, nous avons pu récupérer 69 875 tweets annotés, répartis dans 504 évènements. Afin de se rapprocher au maximum du contexte réel d'un flux de données, nous avons organisé le jeu de données par ordre chronologique de publication. Nous divisons ensuite le jeu de données en deux parties, le jeu d'entraînement et le jeu de test, présentés par la suite. Des détails sur le jeu de données sont présentés dans les figures suivantes.

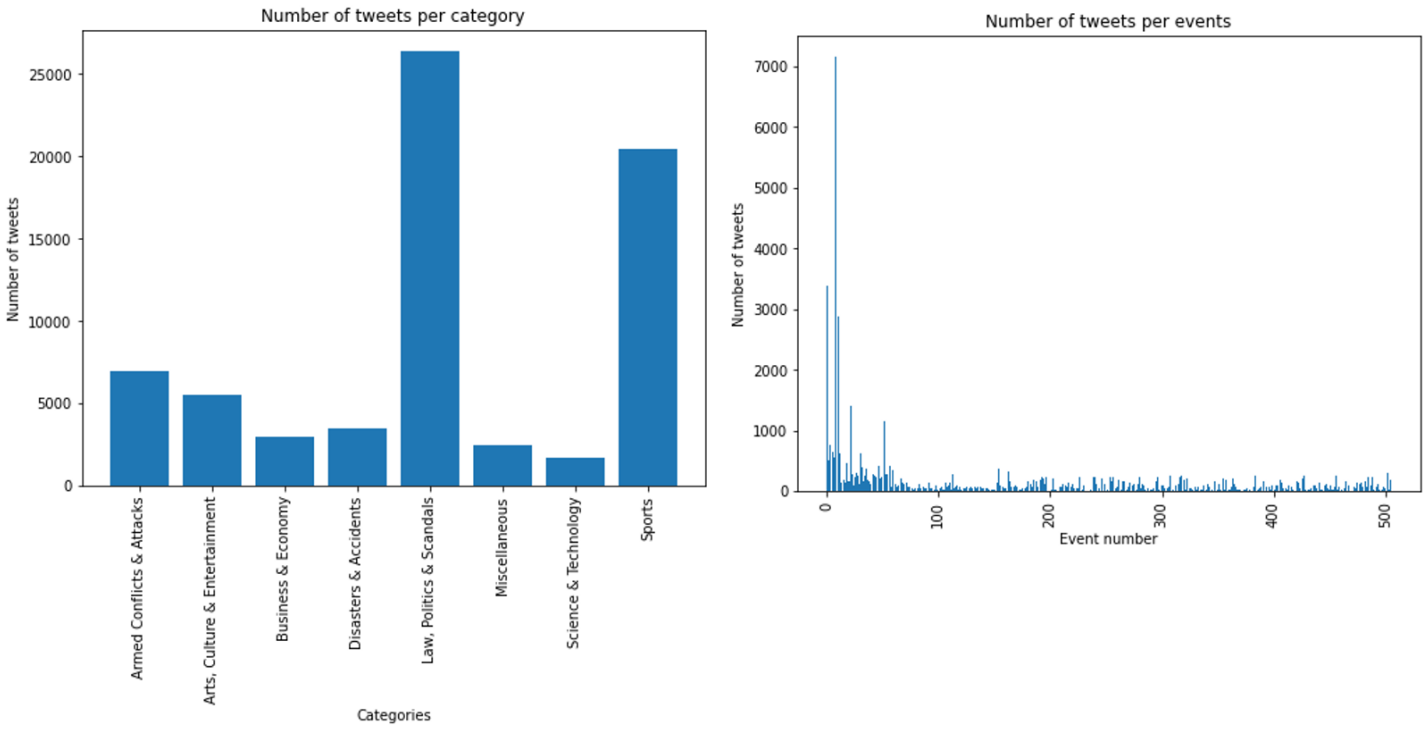

Figure 3. *Nombre de tweets par catégorie et par évènement : la répartition n'est pas homogène*

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. *Des exemples d’évènements du jeu de données. Nous avons choisi les évènements comportant le plus de tweets.*

Nous présentons dans la section suivante les différentes expériences que nous avons menées, leurs objectifs et les résultats obtenus.

3. Expérimentations et Résultats

Dans cette section, nous présentons tout d'abord les métriques d'évaluation et l'organisation du jeu de données utilisées au cours des expériences. Ensuite, nous présentons les différentes expériences menées au cours de notre étude et les résultats obtenus, avant de proposer une discussion à propos de ces résultats.

3.1. *Configuration expérimentale*

Dans cette partie, nous présentons tout d'abord les mesures d'évaluation utilisées puis comment nous avons découpé le jeu de données.

3.1.1. *Mesures d'évaluation*

Nous utilisons pour mesurer la performance des différents modèles b-cubed, qui est une généralisation de la Précision, du Rappel et du F1-score pour le partitionnement et est la méthode d'évaluation des partitionnements la plus complète [AMI 09]. La précision *P* est définie comme la proportion de documents dans le partitionnement du document qui correspondent au même évènement. Le rappel *R* est défini comme la proportion de documents qui correspondent au même évènement, qui sont aussi dans le cluster du document. Le F1 score est lui obtenu en utilisant la formule **[1]**. B-cubed est illustré à la Figure 5.

$$F1 = \frac{2 * P * R}{P + R} \qquad [1]$$

3.1.2. Découpe du jeu de données

Pour mener nos expériences sur un jeu de données aussi proche que possible de la réalité, nous organisons les documents par ordre chronologique et les découpons en fenêtres. Il s'agit d'un paramètre particulièrement important pour la phase d'entraînement du modèle S-BERT, que nous détaillons ensuite. En effet, la grande majorité des labels d'événements qui sont présents dans le jeu d'entraînement ne le sont pas dans le jeu de test. L'ensemble d'apprentissage est constitué de 225 événements, tandis que l'ensemble de test est constitué de 303 événements. Il y a 24 événements communs dans ces ensembles. Ceci est illustré dans la figure 6.

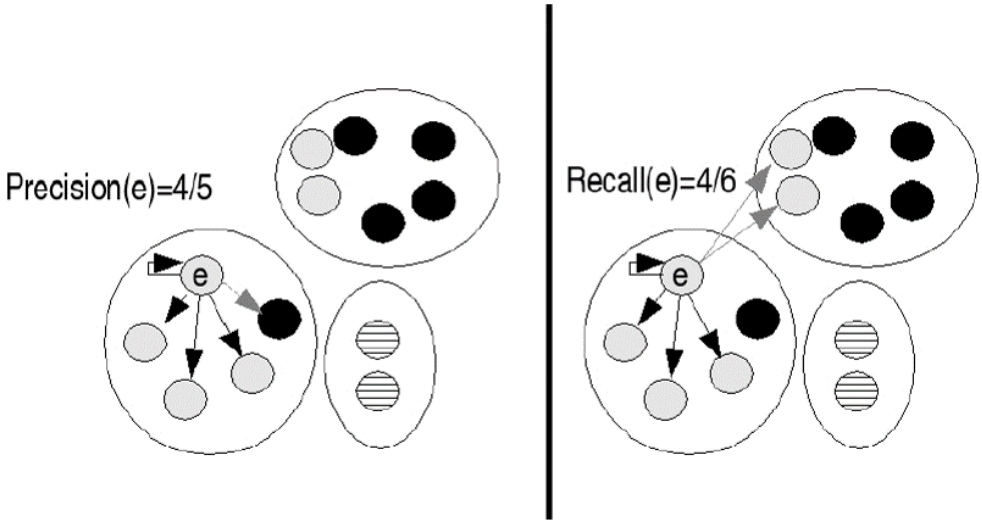

Figure 5. *Un exemple de la manière dont sont calculés la précision et le rappel pour un élément. Concernant l'exemple de la précision, le regroupement contient 4 éléments gris et un élément noir. Ainsi, la précision des éléments gris est de 4/5 tandis que la précision de l'élément noir est de 1/5. Pour obtenir la précision totale, une moyenne est calculée sur l'ensemble des regroupements, en choisissant la précision maximale pour chacun d'entre eux. Dans l'exemple présent, la précision retenue sera celle des éléments gris. Ce calcul est reproduit de la même manière pour le rappel. Figure extraite de [AMI 09]*

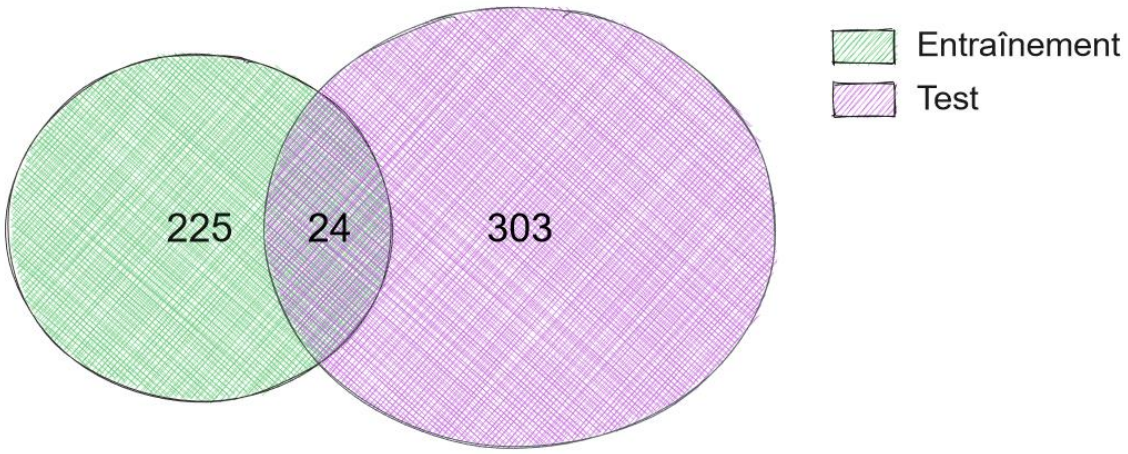

Figure 6. *Répartition des événements entre l'ensemble d'apprentissage et l'ensemble de test. Seuls quelques événements sont communs, en raison de la dérive des conversations.*

3.2. *Modèles de représentation*

Dans cette partie, nous détaillons les différents modèles de représentations textuels utilisés dans les expériences qui suivent.

Nous proposons deux variations de TF-IDF et de S-BERT, tandis que nous utilisons le modèle USE-large[1], que nous appellerons **USE**. Concernant TF-IDF, nous utilisons les implémentations proposées par [MAZ 20][2]. La première, que nous appellerons dans le reste de ce papier **TF-IDF dataset**, propose un IDF calculé sur les tweets labélisés du jeu de données. La seconde, **TF-IDF all tweets**, a un IDF calculé sur l'ensemble du jeu de données. Concernant S-BERT, la première version, nommée **S-BERT nli** est la version pré-entraînée sur le dataset NLI disponible dans les implémentations proposées par les auteurs du papier [REI 19][3]. Ainsi, ce modèle est un réseau siamois, composé de deux modèles BERT égaux. Ces modèles ont été affinés sur le dataset SNLI discuté dans l'état de l'art. Nous avons choisi ce modèle de BERT notamment car le dataset NLI est connu pour améliorer les performances sur les tâches de clustering [BOW 15]. Pour la seconde version, **S-BERT fine-tuned**, nous avons réalisé un affinage du modèle S-BERT sur le jeu d'entraînement, qui constitue la première moitié du jeu de données. Les évènements ont été utilisés comme labels d'entraînement. La particularité de cet entraînement vient de l'organisation temporelle du jeu de données : la majeure partie des évènements présents dans la collection de test ne le sont pas dans la collection d'entraînement, comme expliqué en partie 3.3. L'affinage a donc été réalisé sur 36 000 tweets, de manière à ce que la découpe jeu d'entraînement/jeu de test corresponde aux fenêtres de tweets que nous avons établies. Nous avons assigné à chaque tweet une paire de tweets : un tweet issu du même label et un tweet issu d'un label différent, conformément aux méthodes d'entraînement classiques des réseaux sociaux siamois. Chacun de ces deux tweets est choisi de manière aléatoire dans le jeu d'entraînement, selon les règles concernant les labels que nous venons d'énoncer.

3.3. *Expérience préliminaire*

3.3.2. *Protocole expérimental*

L'objectif de cette expérience est de comparer les différentes manières de discrétiser le flux. Il s'agit d'un paramètre important de notre modèle car il a un impact direct sur les résultats du partitionnement. Notre intuition est que les documents parlant des mêmes événements sont postés dans un court laps de

1:https://tfhub.dev/google/universal-sentence-encoder-large/5

2:https://github.com/ina-foss/twembeddings

3:https://github.com/UKPLab/sentence-transformers

temps. Par la suite, un suivi inter-fenêtre de l'évolution des évènements détectés pourra être fait, comme présenté dans [FED 19] ou [MAI 22]. Nous comparons les performances de différents modèles de représentation de texte avec différentes valeurs de fenêtres dans le contexte de EDF. Nous expérimentons deux types de fenêtres : les fenêtres temporelles et les fenêtres à nombre fixe de documents. Chaque type de fenêtre a ses avantages : la fenêtre temporelle fixe permet de s'assurer que les documents sont postés dans une période de temps courte. Cependant, certaines fenêtres peuvent être presque vides en raison de la variation du nombre de tweets dans la journée, rendant les événements détectés potentiellement non-pertinents. En ce qui concerne les fenêtres de taille fixe, elles permettent d'anticiper l'utilisation de la mémoire de l'algorithme. Cela peut être important dans certains cas, lorsque l'algorithme fonctionne sur une machine avec des ressources mémoire limitées.

Dans cette expérience, nous utilisons 6 fenêtres différentes et comparons les résultats. Nous utilisons des fenêtres de temps de 1 heure, 2 heures et 4 heures. Nous utilisons des fenêtres de taille fixe de 1000 tweets, 2000 tweets, et 4000 tweets. Les tailles de ces fenêtres ont été choisies de manière à ce que la taille soit représentative tout en étant faiblement étalée dans le temps.

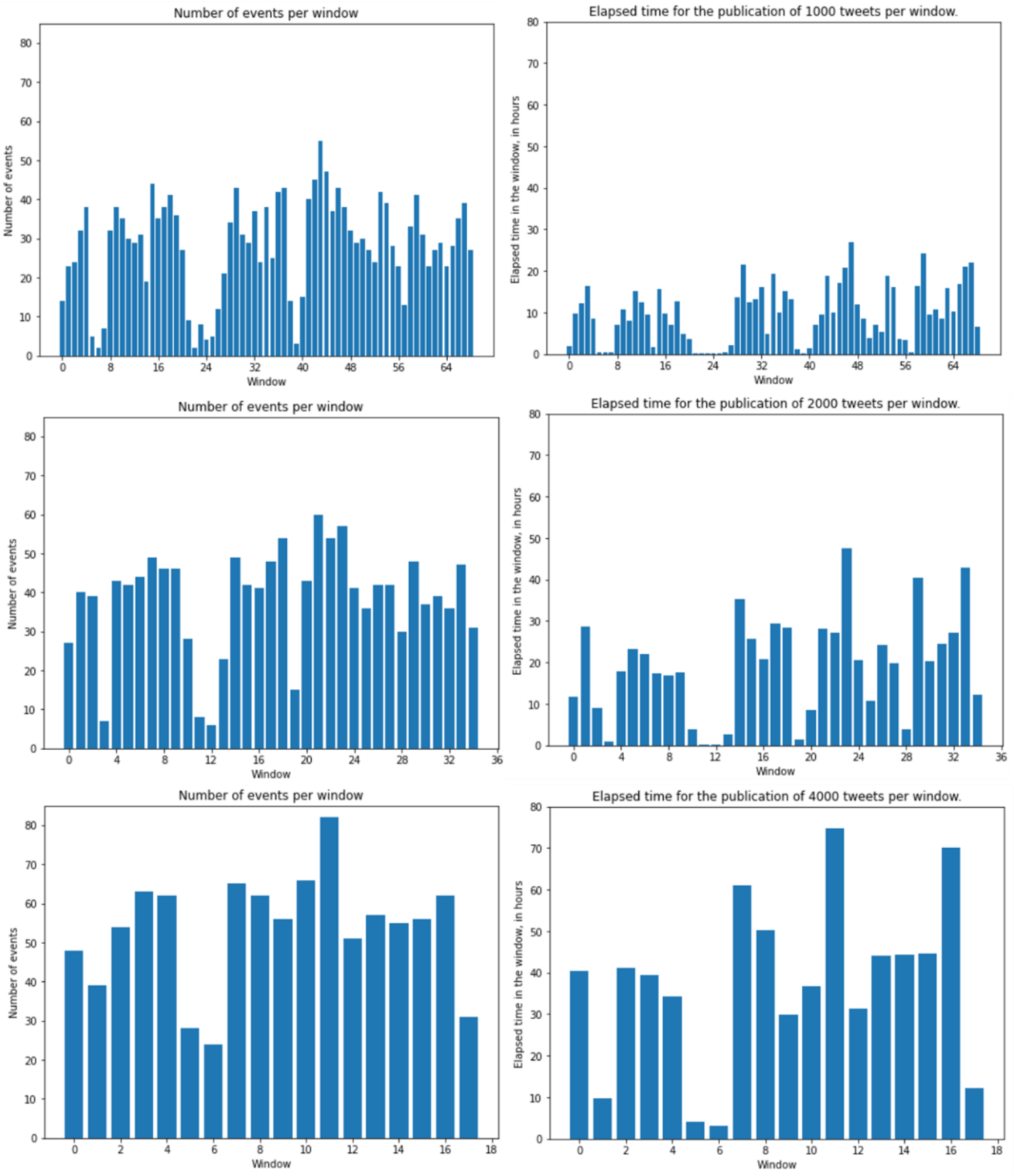

Figure 7. *Nombre d'évènements par fenêtre, en fonction de la taille de la fenêtre. Les fenêtres sont numérotées successivements. Comme nous pouvons le constater, quelque soit le nombre de tweets par fenêtre, le nombre d'évènements varie fortement d'une fenêtre à l'autre. Cependant, nous pouvons observer des caractéristiques similaires lorsque le nombre de tweets varie.*

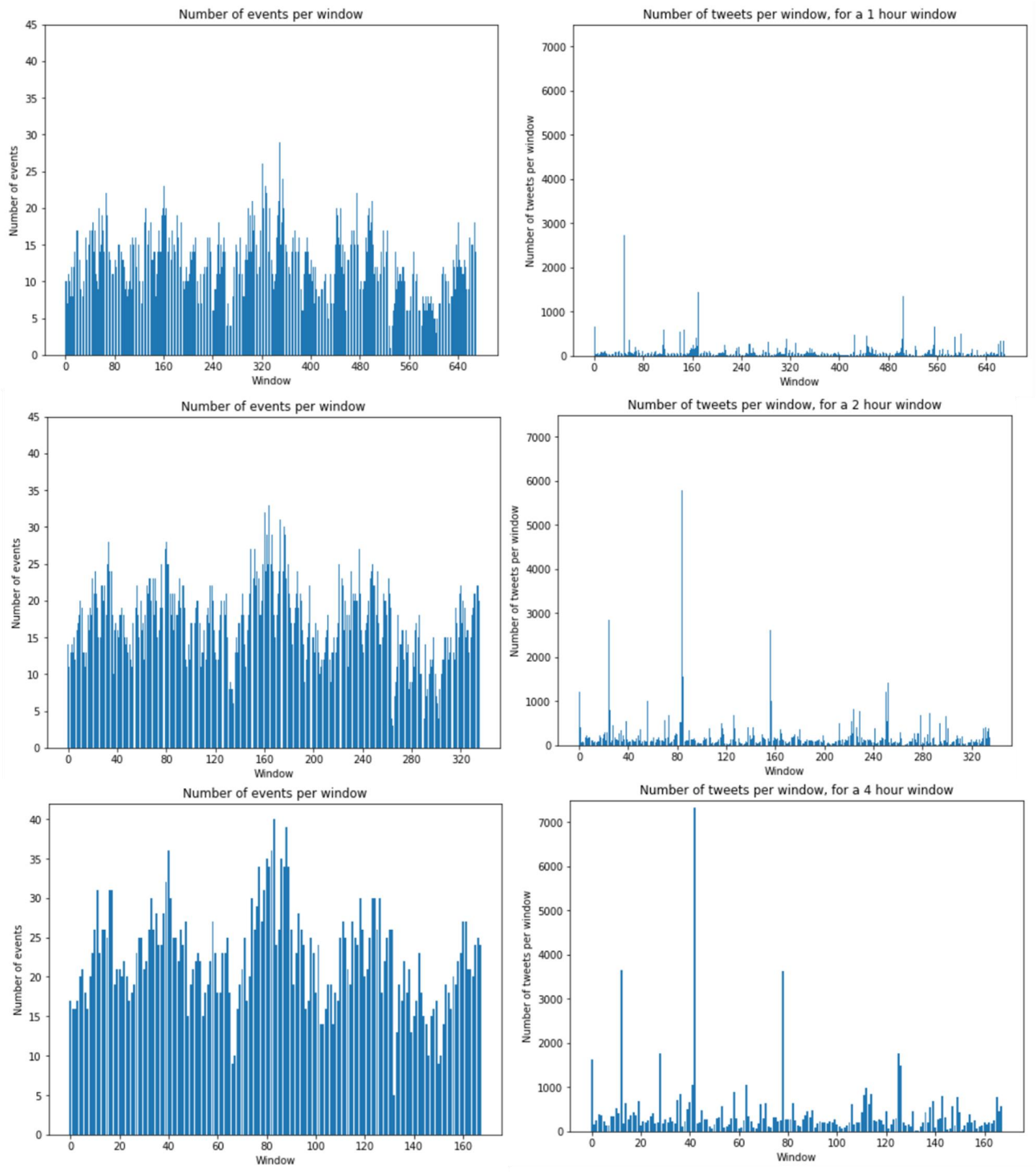

Figure 8. *Nombre d'événements par fenêtre temporelle et nombre de tweets par fenêtre temporelle. Comme nous pouvons le voir, il y a beaucoup de disparité entre les fenêtres. Cependant, nous pouvons observer des caractéristiques similaires lorsque le temps écoulé varie.*

Fenêtre	Modèle	Précision	Rappel	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 heure	TF-IDF dataset	**0.84** ± 0.09	0.80 ± 0.21	**0.80** ± 0.13
	TF-IDF all tweets	**0.84** ± 0.09	0.80 ± 0.21	**0.80** ± 0.13
	USE	**0.91** ± 0.08	0.86 ± 0.19	**0.87** ± 0.12
	SBERT	**0.97** ± 0.05	**0.56** ± 0.22	**0.68** ± 0.19
2 heures	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	**0.87** ± 0.17	0.86 ± 0.11
	SBERT	0.96 ± 0.05	0.51 ± 0.19	0.64 ± 0.18
4 heures	TF-IDF dataset	0.80 ± 0.08	**0.84** ± 0.19	0.80 ± 0.12
	TF-IDF all tweets	0.80 ± 0.08	**0.84** ± 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

Tableau 3.1. *La qualité du partitionnement selon la métrique B-cubed pour chaque représentation textuelle, selon le type de fenêtre. Les fenêtres temporelles semblent les plus adaptées, puisque pour chacune des métriques et pour chaque modèle de représentation, les meilleurs résultats sont obtenus avec ces fenêtres.*

3.3.2. *Résultats*

Les résultats sont présentés dans le tableau 3.1 et différentes analyses des fenêtres sont présentées dans les figures 7 et 8 afin de mieux comprendre la répartition des évènements ainsi que des tweets selon la taille des fenêtres. Nous commençons par l'analyse des fenêtres de taille fixe. Comme nous pouvons le constater, le score F1 est relativement stable lorsque le nombre de tweets par fenêtre varie. Cependant, les valeurs de précision et de rappel ne sont pas stables : lorsque le nombre de tweets par fenêtre augmente, la précision diminue alors que le rappel augmente. En travaillant avec des fenêtres temporelles, les résultats sont vraiment stables. Ceci est probablement dû au fait qu'il y a moins de différences induites par la variation de la durée de chaque fenêtre que par la variation du nombre de tweets par fenêtre. Les tweets labellisés ne sont pas distribués de manière égale dans le temps et une variation d'un millier de tweets est proportionnellement grande par rapport à la taille du jeu de données.

Globalement, les performances sont meilleures lorsque le flux est discrétisé en utilisant des fenêtres de temps. Même si cela implique que chaque fenêtre aura un nombre différent de tweets et occupera donc un espace différent en mémoire, nous pensons qu'il s'agit d'une meilleure méthode de discrétisation, qui de plus est plus cohérente avec ce qui est habituellement fait dans le domaine du

journalisme et de l'information. En effet, il est habituel d'entendre qu'il y a une mise à jour des nouvelles toutes les heures lorsque quelque chose se produit. Puisque les différences entre les valeurs temporelles ne sont pas significatives, nous effectuerons nos prochaines expériences en utilisant des fenêtres temporelles d'une heure, car nous pensons que c'est une meilleure granularité pour comprendre le déroulement des événements.

3.4. *Première expérience*

3.4.1. *Protocole expérimental*

La première expérience est la comparaison des 4 modèles de représentation de texte, **TF-IDF dataset**, **TF-IDF all tweets**, **S-BERT nli** et **USE** dans deux contextes différents : avec application de l'algorithme FSD ou avec application de notre solution EDF. Nous utilisons l'implémentation de FSD proposée par [MAZ 20][4], en adaptant cette solution. En effet, contrairement à l'implémentation proposée, nous utilisons comme mesure de performance la mesure B-Cubed. Concernant les valeurs seuils de l'algorithme FSD proposé, nous avons utilisé les valeurs présentées par [MAZ 20], c'est-à-dire t=0.65 pour TF-IDF dataset, t=0.75 pour TD-IDF all tweets et t=0,39 pour S-BERT et t=0,22 pour USE. Les valeurs de seuils utilisées pour la suppression des arêtes du graphe dans l'approche EDF sont les suivantes : t=0,10 pour les modèles basés sur TF-IDF, t=0,80 pour le S-BERT, t=0,40 pour USE. Pour rappel, les valeurs correspondent à des similarités calculées via Cosine Similarity. Ces valeurs seuils ont été déterminées empiriquement, en choisissant celles maximisant les performances de chacun des modèles.

3.4.2. *Résultats*

Le *Tableau 3.2* résume les résultats. Les chiffres présentés pour EDF sont les moyennes sur l'ensemble des fenêtres de chaque métrique ainsi que l'écart-type. Pour l'algorithme FSD, du fait de la nature de l'algorithme, nous n'avons pas utilisé de fenêtres et donc nous obtenons uniquement une valeur pour chaque métrique. Nous pouvons voir que pour chacune des métriques et chacun des modèles de représentation, EDF est plus performant que le FSD, et que cette observation est particulièrement vraie pour la mesure de rappel.

Modèle	Partitionnement	Précision	Rappel	F1 Score
TF-IDF dataset	FSD	0,70	0,59	0,64
	EDF	**0.84** ± 0.09	**0.80** ± 0.21	**0.80** ± 0.13
TD-IDF all tweets	FSD	0,82	0,50	0,62
	EDF	**0.84** ± 0.09	**0.80** ± 0.21	**0.80** ± 0.13
USE	FSD	0,85	0,38	0,52
	EDF	**0.91** ± 0.08	**0.86** ± 0.19	**0.87** ± 0.12
S-BERT-nli	FSD	0,95	0,31	0,46
	EDF	**0.97** ± 0.05	**0.56** ± 0.22	**0.68** ± 0.19

Tableau 3.2. *Qualité des clusters créés selon la métrique B-cubed, pour chacune des représentations textuelles, en fonction de l'algorithme de partitionnement. L'approche EDF a quasi-systématiquement les meilleurs résultats. Les résultats de EDF sont accompagnés d'une valeur ± parce que la valeur est la moyenne sur l'ensemble des fenêtres. FSD est une seule évaluation.*

4 https://github.com/ina-foss/twembeddings

3.5. *Seconde expérience*

3.5.1. *Protocole expérimental*

La seconde expérience consiste à comparer les performances de **TF-IDF dataset**, **TF-IDF all tweets**, **S-BERT nli** et **USE** dans le contexte EDF. Cette expérience nous sert à comparer les approches de représentation du texte entre elles, de manière à déterminer quelle solution de représentation des données textuelles est la plus efficace, en particulier, nous souhaitons examiner les performances des approches basées sur les Transformers et les comparer aux performances des modèles basés sur TF-IDF, reconnus comme étant plus performants, notamment dans [MAZ 20]. Les performances sont évaluées à l'aide de la métrique B-cubed. Nous formulons l'hypothèse H0 suivante : "Aucune des approches n'est significativement meilleure que les autres". Les valeurs de seuils utilisées pour cette expérience sont les mêmes que précédemment, c'est-à-dire t=0,10 pour les modèles basés sur TF-IDF, t=0,80 pour S-BERT et t=0,40 pour USE.

3.5.2. *Résultats*

Nous comparons chacune des méthodes en les appliquant à chacune des fenêtres définies précédemment, dans un contexte non-supervisé puisqu'aucun des modèles n'a nécessité de label pour un entraînement. Les résultats obtenus sont ceux présentés dans le *Tableau 3.2*, sur les lignes correspondant à l'approche EDF. Les résultats des tests de significativité sont présentés dans le *Tableau 3.3*.

	Précision	**Rappel**	**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

Tableau 3.3. *P-value pour le test Wilcoxon signed-rank. Dans chacun des cas, P-value<α.*

3.6. *Troisième expérience*

3.6.1. *Protocole expérimental*

Pour cette expérience, nous complétons la structure par quelques expériences supplémentaires afin de faciliter la compréhension de l'intuition derrière l'objectif de l'expérience. Bien qu'il puisse sembler évident que l'affinage d'un modèle de langue basé pour une tâche améliorera les performances du modèle, ce n'est pas nécessairement le cas dans un contexte de dérive conceptuelle. Comme nous l'avons vu précédemment, la plupart des événements cibles des ensembles de test n'existent pas dans l'ensemble d'apprentissage. Ainsi, si le modèle apprend réellement de la phase de formation, on pourrait comprendre que le modèle est capable de généraliser certaines des informations qu'il a apprises pendant cette phase de formation. Avant de réaliser l'expérience proprement dite, nous avons réalisé une première expérience où nous avons divisé le jeu de données en deux, sans tenir compte de l'ordre de publication des documents. Ensuite, nous avons visualisé la représentation obtenue pour l'ensemble de test, en utilisant la méthode t-SNE. Nous montrons les résultats de cette expérience dans la figure 9. Ensuite, nous avons réalisé la même expérience en utilisant cette fois-ci le jeu de données ordonné dans le temps. Même si les performances sont moins bonnes, il y a toujours une amélioration notable par rapport à la configuration sans affinage. Nous avons donc décidé de mener l'expérience suivante.

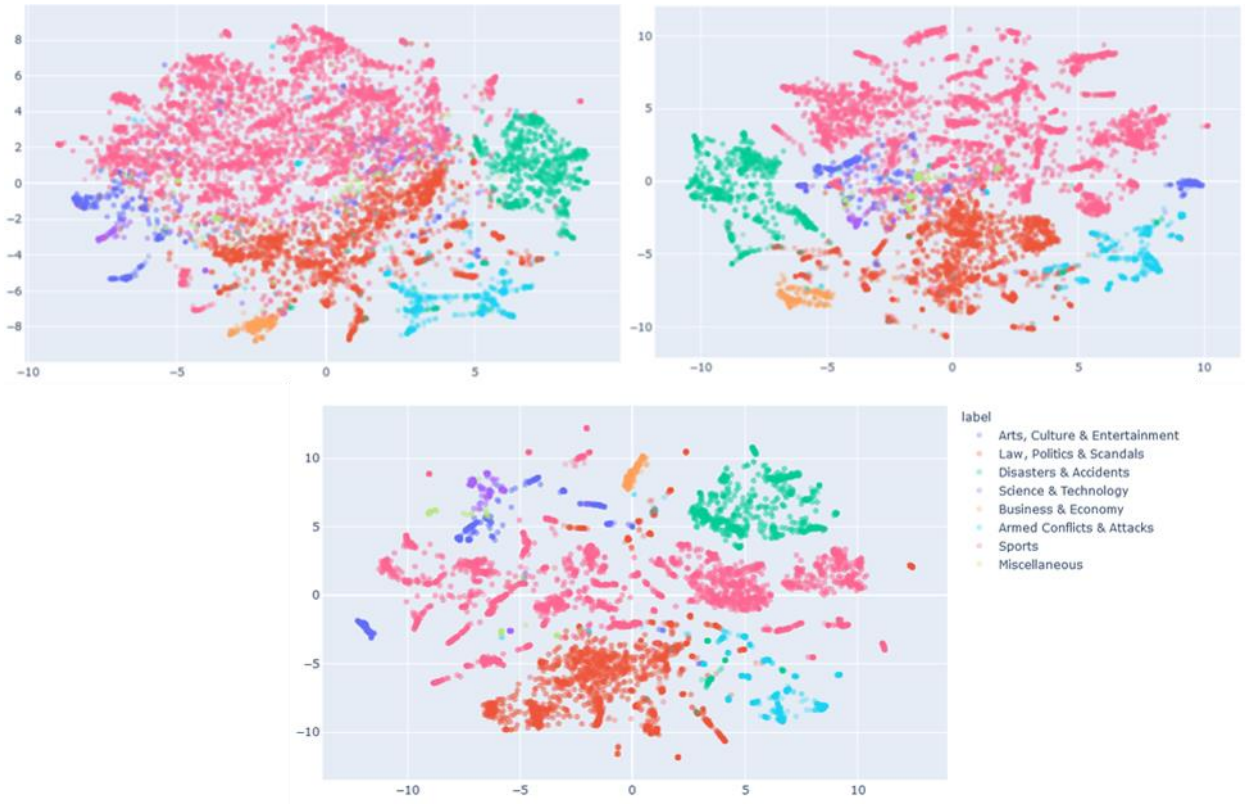

Figure 9. *Représentation t-SNE des plongements S-BERT des documents de l'ensemble de test, dans trois configurations, de gauche à droite et de haut en bas : sans affinage, avec affinage sur l'ensemble ordonné dans le temps, avec affinage sur la moitié de l'ensemble de données, choisi aléatoirement. Comme nous pouvons le voir, même si les groupes de documents semblent être approximativement regroupés par catégorie dans l'image en haut à gauche, il ne semble pas qu'ils créent clairement des clusters différents pour chaque événement. Dans les deux autres images, les clusters semblent plus évidents et pourraient correspondre à des événements. Comme on peut s'y attendre, l'entraînement sur des données aléatoires est plus efficace que l'entraînement sur des données ordonnées dans le temps, car l'ensemble d'entraînement est plus représentatif de ce qui sera rencontré sur l'ensemble de test. Cependant, il ne s'agit pas d'un scénario réaliste. L'entraînement sur l'ensemble d'entraînement de manière ordonnée dans le temps semble toujours être bénéfique, ce qui explique pourquoi nous avons décidé de mener cette expérience.*

3.6.2. *Résultats*

Les résultats sont présentés dans le *Tableau 3.4*. Les résultats des tests de significativité sont présentés dans le *Tableau 3.5*.

Nous pouvons constater que les résultats sont significativement meilleurs pour les architectures Transformers comparé aux approches TF-IDF. En particulier, S-BERT est plus performant en termes de précision (0.95 > 0.83) et de F1-score (0.83 > 0.79) tandis que USE est plus performant selon l'ensemble des métriques.

Modèle	Précision	Rappel	F1 Score
TF-IDF dataset	0.83 ± 0.10	0.79 ± 0.20	0.79 ± 0.13
TD-IDF all tweets	0.83 ± 0.10	0.79 ± 0.20	0.79 ± 0.13
USE	0.95 ± 0.06	0.77 ± 0.17	0.83 ± 0.12
S-BERT-nli	0.90 ± 0.08	0.86 ± 0.18	0.86 ± 0.12

Tableau 3.4. *Qualité des clusters créés selon la métrique B-cubed, pour chacune des représentations textuelles, dans un contexte supervisé, sur le jeu de test.*

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

Tableau 3.5. *P-value pour le test Wilcoxon signed-rank. Etant donné que les résultats des deux méthodes IDF sont similaires, nous considérons que les valeurs des tests de significativité sont les mêmes.*

3.6. *Discussion générale des résultats obtenus*

L'expérience 1 nous a permis de montrer que notre approche EDF est supérieure à l'approche FSD dans le contexte présenté. Ce constat est particulièrement vrai pour la mesure du rappel. Concernant la précision, notamment pour les architectures Transformers, les valeurs entre FSD et EDF sont proches. Nous pensons que l'algorithme FSD permet, dans ces cas-là, d'obtenir des partitionnements cohérents (forte précision). Dans un même temps, FSD a tendance à segmenter les documents d'un même label dans plusieurs partitionnements entraînant une chute importante du rappel. Cela est probablement dû au fait que les partitionnements avec FSD peuvent être créés à l'arrivée d'un nouveau document sans tenir compte de la totalité des documents de la fenêtre. Cette segmentation est moins présente dans la méthode EDF conduisant à une meilleure valeur de rappel.

Nous avons également montré que les approches basées sur des architectures Transformers, particulièrement USE et S-BERT affiné, sont compétitives par rapport aux approches classiques (TF-IDF). Les performances de USE sont particulièrement intéressantes, car ce modèle obtient des résultats similaires à TF-IDF, sans le moindre entraînement sur les données du corpus, laissant percevoir des capacités de généralisation et une bonne adaptabilité à de nouvelles données. Il est notable que S-BERT a des performances moindres que USE. Nous pensons que cela peut être expliqué par les données utilisées pour pré-entraîner les différents modèles Transformers. En effet, le modèle S-BERT que nous avons utilisé est basé sur BERT NLI, qui est entraîné sur le corpus Wikipédia anglais, BookCorpus et affiné sur SNLI. USE quant à lui est entraîné sur un panel de données plus diversifié, incluant des données de forums de discussion ou de sites de questions-réponses, plus proche dans leur formulation (moins formel) des données de Twitter que ne l'est le jeu d'entraînement de S-BERT. De ce fait, les données issues des réseaux sociaux, dont la syntaxe est très particulière notamment du fait de la déstructuration de la langue utilisée (français, anglais...), posent des problèmes à S-BERT non affiné car entraîné sur des données écrites dans un Anglais "plus conventionnel". Une fois le modèle S-BERT affiné sur des données issues de réseaux sociaux, les performances de S-BERT augmentent et deviennent comparables aux autres modèles. Ainsi, nous pouvons souligner l'importance de la phase d'affinage du modèle et l'intérêt que pourrait représenter un pré-entraînement de S-BERT directement sur des données issues des réseaux sociaux pour obtenir de meilleurs résultats dans notre contexte.

Conclusion

Nous avons étudié le problème de la détection d'évènements dans les flux de données textuelles sous la forme d'une tâche de partitionnement. Dans un premier temps, nous avons montré la supériorité de notre approche EDF basée sur du partitionnement par rapport à l'approche FSD dans le cadre de fenêtres temporelles. Ensuite, nous avons étudié les performances de différents modèles de représentation du texte, dans des contextes supervisés et non supervisés et conclu que les approches TF-IDF ne peuvent pas être déclarées comme supérieures aux approches Transformers, même lorsque celles-ci ne sont pas entraînées sur ce corpus en particulier. Cela ouvre des perspectives intéressantes, notamment du fait de la variation du vocabulaire utilisé sur les réseaux sociaux, qui limitait les approches telles que TF-IDF jusqu'à maintenant. Aussi, d'un point de vue apprentissage, des approches d'apprentissage actif ou incrémental peuvent être considérées pour un potentiel développement de ces approches dans le contexte des flux de données issus des réseaux sociaux. En effet, cela permettrait d'adapter les Transformers, plus performants lorsque affinés, à des contextes où il est difficile d'avoir

accès à des données labellisées. Afin de compléter les travaux présentés, il serait intéressant de faire varier les sources de données (par exemple en utilisant des unes de journaux) ou encore la langue utilisée dans les données. Il sera aussi intéressant de mener une comparaison de différentes approches de partitionnement, à la manière de ce qui a été fait ici pour les représentations textuelles. Enfin, ce papier s'inscrit dans un projet de recherche mené en collaboration avec l'entreprise Scalian. L'objectif est de travailler sur une chaîne plus complète [MAI 20] de détection, de suivi et de caractérisation des évènements, à la manière de la méthode présentées dans [FED 19], notamment pour faire un suivi inter-fenêtre des évènements et déterminer les principaux composants de ces évènements. l sera alors possible d'avoir une description de l'évènement selon différentes granularités temporelles, c'est-à-dire à l'échelle de la fenêtre ou un suivi plus général.

Bibliographie

[AMI 09] AMIGÓ E., GONZALO J., ARTILES J. AND VERDEJO F., « A comparison of extrinsic clustering evaluation metrics based on formal constraints », *Information retrieval*, 12(4) :461–486, 2009

[ATE 15] ATEFEH F. and KREICH W., « A survey of techniques for event detection in twitter », *Comput. Intell.*, 31(1):132–164, February 2015.

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

[ALL 00] ALLAN J., LAVRENKO V., MALIN D. AND SWAN R., « Detections, bounds, and timelines : Umass and tdt-3 », *Proceedings of Topic Detection and Tracking Workshop*, 11 2000.

[ALL 17] ALLAHYARI M., POURIYEH S., ASSEFI M., SAFAEI S., TRIPPE E, GUTIERREZ J. AND KOCHUT K. « A brief survey of text mining : Classification, clustering and extraction techniques », 2017.

[AGG 12] AGGARWAL C. AND ZHAI C., « A survey of text clustering algorithms », *Mining text data*, pages 77–128. Springer, 2012.

[BAE 99] BAEZA-YATES R. AND Ribeiro-Neto B., *Modern Information Retrieval*, ACM Press / Addison-Wesley, 1999.

[BOW 15] BOWMAN S., ANGELI G., POTTS C. AND MANNING C. « A large annotated corpus for learning natural language inference», *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

[BOJ 16] BOJANOWSKI P., GRAVE E., Joulin A. and Mikolov T., « Enriching word vectors with subword information », *arXiv preprint*, arXiv :1607.04606, 2016.

[BOO 16] DE BOOM C., VAN CANNEYT S., DEMEESTER T. AND DHOEDT B., « Representation learning for very short texts using weighted word embedding aggregation », *CoRR*, abs/1607.00570, 2016.

[BRO 94] BROMLEY J., GUYON I., LECUN Y, SÄCKINGER E. AND SHAH R, « Signature verification using a" siamese" time delay neural network », *Advances in neural information processing systems*, pages 737–737, 1994.

[BLO 08] BLONDEL V., GUILLAUME J., LAMBIOTTE R. AND LEFEBVRE E., « Fast unfolding of communities in large networks », *Journal of statistical mechanics : theory and experiment*, 2008(10) :P10008, 2008.

[BEC 10] BECKER H., NAAMAN M. AND GRAVANO L., « Learning similarity metrics for event identification in social media », *Proceedings of the third ACM international conference on Web search and data mining*, pages 291–300, 2010.

[BEC 11] BECKER H., NAAMAN M. AND GRAVANO L., « Beyond trending topics : Real-world event identification on twitter », volume 11, 01 2011.

[CON 17] CONNEAU A., KIELA D., SCHWENK H., BARRAULT L., AND BORDES I. « Supervised learning of universal sentence representations from natural language inference data », *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, 2017.

[CAS 11] CASTILLO C., MENDOZA M. and POBLETE B., « Information credibility on twitter », *Proceedings of the 20th international conference on World wide web*, pages 675–684, 2011.

[CER 18] CER D., YANG Y., KONG S., HUA N., LIMTIACO N, ST JOHN R., CONSTANT N., GUAJARDO-CÉSPEDES M., YUAN S., TAR C., SUNG Y., STROPE B, KURZWEIL R., « Universal Sentence Encoder», *arXiv preprint*, arXiv :1803.11175, 2018.

[DEV 18] DEVLIN J., CHANG M., Lee K and Toutanova K. « Bert : Pre-training of deep bidirectional transformers for language understanding », *arXiv preprint*, arXiv :1810.04805, 2018.

[FED 19] FEDORYSZAK M., BRENT A., RAJARAM V. AND ZHONG C., « Detection and resolution of rumours in social media : A survey », *CoRR*, abs/1907.11229, 2019.

[GUI 14] GUILLE A. AND FAVRE C., « Mention-anomaly-based event detection and tracking in twitter », *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 375–382, 2014.

[HAS 18] HASAN M., ORGUN M. and SCHWITTER R., « A survey on real-time event detection from the twitter data stream », *J. Inf. Sci.*, 44(4) :443–463, August 2018.

[HAS 19] HASAN M., ORGUN M. and SCHWITTER R., « Real-time event detection from the twitter data stream using the twitternews+ framework», *Information Processing and Management*, 56(3) :1146–1165, 5 2019.

[JON 72] JONES K. « A statistical interpretation of term specificity and its application in retrieval », *Journal of documentation*, 1972.

[KWA 11] KWAK H., LEE C. and Park H., « What is twitter, a social network or a news media ? », *Proceedings of the 19th international conference on World wide web*, pages 591–600, 2010.

[KIR 15] KIROS R., ZHU Y., SALAKHUTDINOV R., ZEMEL R., TORRALBA A, URTASUN R. AND FILDER S, « Skip-thought vectors », *arXiv preprint*, arXiv :1506.06726, 2015.

[MAI 22] MAITRE E., *Détection d'évènements dans des flux de textes courts pour la prise de décision. (Event detection on stream of short texts for decision-making)*, PhD thesis, Toulouse, 2022.

[MAI 20] MAÎTRE E., CHEMLI Z., CHEVALIER M., DOUSSET B., GITTO J AND TESTE O., « Event detection and time series alignment to improve stock market forecasting », *Proceedings of the First Joint Conference of the Information Retrieval Communities in Europe (CIRCLE 2020)*, volume 2621 of CEUR Workshop Proceedings. CEUR-WS.org, 2020.

[MAI 21] MAITRE E., CHEMLI Z., CHEVALIER M., DOUSSET B., GITTO J AND TESTE O., « Étude de l'in-fluence des représentations textuelles sur la détection d'évènements non supervisée dans des flux de données», *Inforsid*, 2021.

[MIK 13] MIKOLOV T., CHEN K., CORRADO G. AND DEAN J., « Efficient estimation of word representations in vector space », *arXiv preprint*, preprint arXiv :1301.3781, 2013.

[MAZ 20] MAZOYER B., HERVE N., HUDELOT C. AND CAGE J., « Représentations lexicales pour la détection non supervisée d'événements dans un flux de tweets : étude sur des corpus français et anglais», *Extraction et Gestion des connaissances*, Janvier 2020.

[MCM 15] MCMINN A. and JOSE J., « Real-time entity-based event detection for twitter », *International conference of the cross-language evaluation forum for european languages*, volume 5, 2011.

[MCM 13] MCMINN A., MOSHFEGHI Y. AND JOSE J., « Building a large-scale corpus for evaluating event detection on twitter », *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 409–418, 2013.

[NAA 11] NAAMAN M., BECKER H. AND GRAVANO L. « Hip and trendy : Characterizing emerging trends on twitter. » *JASIST*, 62 :902–918, 05 2011.

[PET 10] PETROVIC S., OSBORNE M. AND LAVRENKO V., « Streaming first story detection with application to twitter », *Human Language Technologies : The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, page 181–189, 2010.

[PAN 10] PAN S. and YANG Q., « A survey on transfer learning», *IEEE Transactions on Knowledge and Data Engineering*, 22(10) :1345–1359, 2010.

[REI 19] REIMERS N. and GUREVYCH I., « Sentence-bert :Sentence embeddings using siamese bert-networks», *CoRR*, abs/1908.10084, 2019.

[REP 18] REPP O. and RAMAMPIARO H., « Extracting news events from microblogs », *Journal of Statistics and Management Systems*, 21(4) :695–723, 2018.

[SAK 10] SAKAKI T., OKAZAKI M. and MATSUO Y., « Earthquake shakes twitter users : Real-time event detection by social sensors », pages 851–860, 01 2010.

[VON 18] VON NORDHEIM G., BOCZEK K AND Koppers L., « Sourcing the sources », *Digital Journalism*, 6(7) :807–828, 2018.

[VAS 17] VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L, GOMEZ A., KAISER L., AND POLOSUKHIN I. « Attention is all you need», *arXiv preprint*, arXiv :1706.03762, 2017.

[WEN 11] WENG J. and LEE B., « Event detection in twitter », *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5, 2011.

[ZUB 18] ZUBIAGA A., AKER A., BONTCHEVA K., LIAKATA M. AND PROCTER R., « Detection and resolution of rumours in social media : A survey », 51(2), February 2018.

