Des clusters de tweets aux tags de descriptions : présentation d'un évènement par la caractérisation de ses manifestations

Olivier Gracianne, Anaïs Halftermeyer, Thi-Bich-Hanh Dao

▶ **To cite this version:**

Olivier Gracianne, Anaïs Halftermeyer, Thi-Bich-Hanh Dao. Des clusters de tweets aux tags de descriptions : présentation d'un évènement par la caractérisation de ses manifestations. Conférence Nationale en Intelligence Artificielle 2022 (CNIA 2022), Jun 2022, Saint-Etienne, France. hal-03866299

HAL Id: hal-03866299
https://hal.science/hal-03866299

Submitted on 22 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Des clusters de tweets aux tags de descriptions : présentation d'un évènement par la caractérisation de ses manifestations

Olivier Gracianne[1,2], Anaïs Halftermeyer[1], Thi-Bich-Hanh Dao[1]

[1] Univ. Orléans, INSA Centre Val de Loire, LIFO EA 4022
[2] Atos France

olivier.gracianne@etu.univ-orleans.fr, [anais.halftermeyer, thi-bich-hanh.dao]@univ-orleans.fr

Résumé

Notre travail aborde le problème de caractérisation de sous-évènements dans des tweets par la description de leur groupement. Nous nous appuyons sur la représentation vectorielle de tweets dans un espace de plongement et les clusterisons. Nous proposons deux méthodes pour sélectionner des candidats parmi les mots des tweets pour construire un ensemble de tags de description. À partir de ces tags, nous proposons de construire une description par cluster en utilisant un modèle déclaratif en programmation linéaire en nombres entiers. Les expérimentations sur un jeu de données réelles montrent l'intérêt de notre approche.

Mots-clés

Twitter, plongement de mots/documents, clustering, description de clusters, programmation linéaire en nombre entier

Abstract

Our work deals with the sub-event characterization in tweets problem through the description of their grouping. We leverage vector representation of tweets in embedding space and cluster them. We propose two methods to select descriptor candidates among tweets' words to build a description tag set. From these tags, we propose to build a description per cluster using an integer linear programming declarative model. Experiments on real data show the interest of our approach.

Keywords

Twitter, word/document embedding, clustering, cluster description, integer linear programming

1 Introduction

L'utilisation massive des réseaux sociaux a produit d'énormes jeux de données pouvant être explorés et exploités avec une grande variété d'outils et de techniques. Chacun de ces médias a ses spécificités qui le rend plus ou moins adapté à exploiter pour une tâche donnée. Twitter offre des données qui ont des propriétés intéressantes, notamment pour la caractérisation d'évènements. En particulier, les messages y sont courts et postés peu de temps après l'évènement en lui-même, par une grande variété de sources. Cette plate-forme fournit une quantité conséquente de données disponibles publiquement à travers une API facile d'utilisation [1]. Depuis août 2020, elle est devenue encore plus populaire avec une hausse significative des limites de requêtage pour les académiques.

Cela a permis aux chercheurs de constituer facilement des jeux de données avec de grands ensembles de tweets. Naturellement, le nombre d'approches s'appuyant sur ce type de données a augmenté dans le même temps.

La détection et la description d'évènements sont devenues plus actives alors même qu'il s'agissait de tâches déjà exploitées.

La plate-forme Twitter permet la diffusion de contenus très diversifiés sur des sujets tout autant variés, parmi lesquels les évènements du monde réel. Ainsi ces évènements sont révélés par le prisme du tweet. Toutefois caractériser ces évènements dans le volume et l'hétérogénéité de la masse informationnelle disponible sur ce média implique de devoir détecter des composants de grain plus fin. Ces manifestations mesurables de l'évènement cible se reflètent dans des sous-ensembles de tweets que nous appelons *sous-évènements*.

Dans ces travaux, nous considérons des tweets émergeant autour d'un évènement, annoncé à l'avance ou non, étant publiés dans la fenêtre temporelle de celui-ci, et qu'il est possible de regrouper en plusieurs sous-évènements structurés. Si nous prenons l'exemple d'une tempête, ses sous-évènements constitutifs et a priori détectables pourraient être des vents violents et des crues menant à des voies de communication coupées. Nous émettons l'hypothèse que le clustering du contenu des tweets ayant un lien avec l'évènement permet de mettre en évidence certains de ses sous-évènements. Nous représentons finalement ces derniers avec leur description, qui est un ensemble de tags, attribuée à un cluster de tweets. Ces groupements sont construits avec des messages contenant un mot ou un hashtag les liant à l'évènement.

Dans cet article, nous présentons notre approche bout-en-bout qui détecte et décrit des sous-évènements en tirant avantage d'outils de TAL bien connus. Nos contributions

1. https ://api.twitter.com/2/tweets/search/all est par exemple le point d'accès à cette API permettant de faire une recherche sur tous les tweets indexés par l'API.

sont :

- Le problème de description formulé avec une approche déclarative, en utilisant les plongements des mots des tweets comme descripteurs ;
- Le pipeline de traitement bout-en-bout utilisé ;
- Nous expérimentons notre méthode sur un jeu de données en français que nous avons collecté en ciblant la tempête Alex ayant frappé la France en octobre 2020. Nos résultats montrent la validité de notre méthode pour sélectionner des mots sémantiquement importants par rapport à leur cluster.

L'organisation du papier est la suivante. Nous faisons un tour d'horizon des travaux connexes à notre approche en section 2. Notre approche est présentée dans la section 3 et les expérimentations dans la section 4. Nous présentons nos conclusions en section 5 et discutons des perspectives de notre travail.

2 Travaux connexes

La détection d'évènements sur Twitter et la description de clusters sont deux domaines de recherches actifs et plusieurs approches existent déjà pour ce média. Nous présentons ici les axes de recherches principaux pour les deux.

2.1 Détection d'évènements sur Twitter

Toutes les approches portées à notre connaissance comportent une détection de brusques changements du comportement des ses utilisateurs. Un évènement détectable sur ce média, virtuel ou du monde réel, découle toujours de ce type de variations. Leur nature peut concerner le contenu des tweets, l'utilisation des fonctions "j'aime" et retweet ou encore la dynamique des communautés.
Ces dernières peuvent être représentées par des graphes d'utilisateurs, comme dans l'approche proposée dans [1] qui s'appuie sur des représentations en graphes des communautés d'utilisateurs et permet ainsi de détecter des mouvements de protestation. [15] propose de suivre l'évolution des communautés pour détecter les évènements significatifs pour leurs utilisateurs.

Nous avons choisi de nous orienter vers des méthodes s'appuyant sur le contenu des tweets, dans le but de pouvoir tirer parti de leur représentation en espace sémantique. Nous ne tenons pas compte des caractéristiques sociales de ce réseau.

La détection d'évènements basée sur le contenu des tweets peut être de différentes natures. Quantifier et/ou qualifier numériquement l'intérêt porté aux mots permet d'identifier ceux qui sont utiles à la détection d'évènements. Dans [17], les auteurs proposent un score permettant de traduire un tweet en signal et utilisent des méthodes de traitement du signal pour repérer les mots indiquant des évènements. Les travaux dans [11] se basent sur le traçage des n-grammes. Ceux qui apparaissent souvent obtiennent un score élevé et les tweets dans lesquels ils sont présents sont clusterisés. Les clusters d'évènements sont identifiés grâce à de la connaissance extérieure (base de donnée de tierce partie). Nous nous inscrivons dans la lignée de ces travaux, en tirant en plus parti des représentation de tweet par plongement.

Les approches à base de calcul de distribution de probabilité d'apparition d'un tweet pour un sujet sont nombreuses. Qu'elles se concentrent sur les sujets liés à des évènements [19] ou qu'elles en intègrent de plus généraux [18], la détection de l'évènement consiste à mesurer la brusque hausse de la probabilité d'affecter un tweet à un sujet. Toutefois, ces approches ne prennent pas en compte la sémantique des mots ou des tweets comme nous souhaitons le faire.

En travaillant à partir du flux de données de Twitter, on reçoit des données en continu. Il est possible d'en changer la représentation et de les clusteriser au fur et à mesure pour pouvoir suivre la dynamique des clusters ainsi formés et de détecter les évènements qui correspondent à de brusques croissances de ces clusters ([2], [9], [5]). Les différentes approches font varier les critères d'association d'un tweet à un cluster ou les règles de construction de ces derniers.
Nous nous concentrons sur l'étape de description des clusters pour valider notre approche sur un ensemble de données statique. Nous prévoyons de nous rapprocher de ces travaux de clustering incrémental ultérieurement.

2.2 Description de clusters

Avoir des clusters de données n'est pas suffisant pour récupérer l'information qu'un cluster contient. Il faut la rendre compréhensible par un humain. Une explication est nécessaire pour comprendre rapidement les résultats d'un processus impliquant autant de données et dans le cadre de nos travaux, pour aider à comprendre quel genre d'évènement est en cours et comment il se manifeste.

Dans [13], des concepts sous la forme d'ensembles clos d'éléments sont recherchés et les clusters sont construits autour. Ces concepts servent finalement d'explication. Dans [3], les clusters descriptifs sont construits en même temps que leur description, les données étant définies par un ensemble d'attributs et un ensemble de tags. Les clusters sont calculés à base des attributs et sont décrits en utilisant les tags. Ces approches se basent toutefois sur des vecteurs binaires, représentant un tweet par un vecteur de la taille du vocabulaire où chaque 1 représente un mot présent dans le tweet. Elles ne peuvent pas tirer parti des espaces de plongement sémantique.
Dans [4], les auteurs proposent de décrire les clusters d'une partition en utilisant un ensemble de descripteurs en lien mais distinct de l'ensemble de données. Par exemple, expliquer une communauté d'utilisateurs de Twitter avec des hashtags employés par les utilisateurs de celle-ci. Ils montrent que c'est un problème difficile computationnellement voire impossible à résoudre mais qui peut être relaxé pour le rendre accessible. Nous ne pouvons pas ré-appliquer cette méthode, comme nous cherchons à expliquer les données directement depuis leur contenu.

L'abstraction et l'extraction de résumé sont des approches bénéficiant du développement des approches neuronales. En reformulant des phrases du corpus cible ou en en extrayant les phrases saillantes, des modèles similaires à BERT

[16] génèrent des résumés pour des ensembles de documents. Ce sont toutefois des approches nécessitant d'une part de grands jeux de données annotées dont nous ne disposons pas, et qui sont encore difficilement explicables d'autre part.

Nouveauté. La contribution de nos travaux réside d'une part dans le fait d'utiliser le contenu même des tweets clusterisés dans un espace sémantique riche pour les décrire et d'autre part dans la stratégie mise en place pour choisir les descripteurs potentiels. Comme nous allons le détailler, nous proposons une méthode de sélection de candidats descripteurs basée sur une mesure de leur intérêt descriptif et une autre intégrant en plus un critère de couverture de l'espace de représentation sémantique du cluster décrit.

3 Approche proposée

Nous cherchons à caractériser un "grand" évènement en décrivant ses différents sous-évènement qui se manifestent sur Twitter. Pour ce faire, nous nous appuyons sur l'hypothèse distributionnelle [8] et son application à travers les plongements de documents [10]. Nous postulons que des tweets traitant d'un même sujet seront regroupés par un algorithme de clustering s'ils sont représentés dans un espace de plongement sémantique. En effet, selon cette hypothèse, les tweets traitant d'un même sujet ont un sens similaire et y sont plus proches. Nous présentons finalement les évènements par la mosaïque des descriptions de leurs sous-évènements que nous construisons.
Dans ce qui suit, nous utiliserons $\mathcal{C}$ pour désigner un clustering (une partition), C_i pour le i-ème cluster, w pour un descripteur (un mot) et d pour une description (un ensemble de mots).

3.1 Représentation des données et clustering

Représentation des données. Pour construire des vecteurs de documents avec les tweets, nous utilisons le plongement proposé dans [10] avec le modèle Doc2Vec. Cette approche permet de construire un espace de représentation sémantiquement fin et basé sur le discours en situation. La distance entre deux vecteurs y représente bien une forme de dissimilarité sémantique entre les tweets correspondants. Par ailleurs, ce modèle permet aussi de contrôler la dimension des vecteurs obtenus. En effet, leur taille a un grand impact à la fois sur la mesure de distance entre les vecteurs et sur le temps de calcul du clustering. Nous avons fixé la taille des vecteurs de documents représentant les tweets à 500, nombre déterminé empiriquement.

Clustering. Selon l'hypothèse distributionnelle [8], l'espace de représentation dans lequel nous plongeons les tweets est riche car directement capté au travers des usages qui sont faits des mots eux-mêmes. Par sa construction, la distance entre deux vecteurs de tweet y représente une forme de distance sémantique entre ces tweets. Nous la mesurons avec la distance euclidienne. Pour construire des clusters sémantiquement cohérents représentant des sous-évènements, nous proposons donc d'utiliser l'algorithme K-Means. En effet, ce dernier construit des clusters compacts au sein de l'espace de représentation dans lequel il travaille en minimisant la distance d'un point au centroïde de son cluster. Nous cherchons ainsi à obtenir les clusters les plus cohérents et centrés sur un objet.

3.2 Descriptions

Nous construisons une description pour un cluster de tweets dans le but de l'interpréter et d'avoir une image de ce qu'il concerne. Nous proposons de considérer les mots des tweets d'un cluster comme ses possibles tags de descriptions et de construire un modèle en Programmation Linéaire en Nombre Entier (PLNE) pour sélectionner les meilleurs mots pour décrire chaque cluster. Pour ce faire nous devons définir un critère indiquant si un mot est adapté pour décrire un cluster et les règles adéquates pour construire une description.

3.2.1 Score DF-IDF

Nous proposons de mesurer la pertinence d'un mot w par rapport à un cluster C_i par le score DF-IDF défini comme suit :

$$DFIDF(w, C_i) = \frac{N_w(C_i)}{|C_i|} \cdot log(\frac{N(\mathcal{C})}{N_w(\mathcal{C})}) \qquad (1)$$

où $N_w(C_i)$ est le nombre de tweets contenant w dans le cluster C_i, $|C_i|$ la taille de C_i, $N(\mathcal{C})$ le nombre total de tweets et $N_w(\mathcal{C})$ le nombre de tweets contenant w dans le clustering $\mathcal{C}$.
Le calcul de ce score est inspiré de [17]. Dans ces travaux, le DF-IDF donne des informations sur l'importance d'un mot pour discriminer le tweet dans lequel il apparaît des autres tweets, sur la base d'une fenêtre temporelle d'apparition (représentée dans la seconde opérande de la formule, l'IDF). Ici, nous en avons adapté la formule pour que le score aide à discriminer le tweet contenant le mot cible sur la base des clusters obtenus. Comme on peut le voir dans la formule 1, plus un terme est fréquent dans un cluster, plus élevé est son DF-IDF. Mais s'il est fréquent dans tout le corpus, son score diminue. L'idée de ce score est d'y compresser une forme de typicalité et de fréquence.

3.2.2 Sélection des descripteurs candidats

La sélection des mots candidats pour décrire des clusters est un point clef de notre approche. Nous présentons ici trois méthodes de sélection.

Plus hautes fréquences. Les 20 mots les plus fréquents dans chaque cluster sont sélectionnés pour constituer la liste initiale des candidats. On en fait l'ensemble final en supprimant leurs occurrences multiples.

Top DF-IDF. Les 20 mots avec les plus hauts scores DF-IDF sont sélectionnés depuis chaque cluster. Nous obtenons l'ensemble de candidats avec le même dédoublonnage. Avec cette méthode les mots sélectionnés sont les plus pertinents d'après le score DF-IDF.

Hybride DF-IDF/FPF. Cette méthode vise à sélectionner les mots pertinents d'après le score DF-IDF qui couvrent le plus la représentation des mots. Pour assurer cette couverture, nous utilisons l'algorithme Farthest Point First (FPF)

[7] qui vise à sélectionner des représentants (têtes) des points d'un ensemble des points. La première tête est le point le plus éloigné de tous les autres et tous les autres points ont ce point comme tête. À chaque itération, le point le plus éloigné de sa tête est sélectionné pour devenir une nouvelle tête. Chaque point plus proche de celle-ci que de sa précédente tête lui est alors raccroché.
Dans notre espace, les points représentent des mots et la distance représente l'éloignement sémantique. Par son fonctionnement, cet algorithme va donc choisir des mots qui en représentent d'autres au sens proche et ces représentants seront sémantiquement distants les uns des autres. Cet algorithme est appliqué dans chaque cluster pour sélectionner 20 têtes, en restreignant les mots possibles à ceux ayant un score DF-IDF au dessus de la médiane de leur cluster. Nous nous assurons ainsi de garantir une certaine pertinence des mots que peut proposer cette sélection hybride.

3.2.3 Construction des descriptions

Nous utilisons la PLNE pour formuler notre problème d'une manière déclarative. Étant donné un clustering $\mathcal{C} = \{C_1, C_2, ..., C_k\}$ et un ensemble de candidats descripteurs $W = \{w_1, w_2, ..., w_m\}$, le modèle prend en entrée :

- une matrice P de taille $k \times m$, où P_{ij} est le score DF-IDF du mot w_j dans le cluster C_i ;
- k matrices T^k de taille $|C_k| \times m$, où T^k_{ij} vaut 1 si le candidat w_j est présent dans le tweet i du cluster C_k.

Nous définissons comme variables une matrice binaire X de taille $k \times m$, où X_{ij} vaut 1 lorsque le mot w_j est utilisé pour décrire le cluster C_i. L'objectif du modèle est de maximiser la pertinence des descriptions construites, ce qui se traduit par la maximisation de la somme :

$$\sum_{i=1}^{k}\sum_{j=1}^{m} X_{ij}P_{ij} \tag{2}$$

Les exigences pour les descriptions sont exprimées par des contraintes du modèle :

- ***Non-vide*** Une description ne doit pas être vide :

$$\forall i \in [1,k], \sum_{j=1}^{m} X_{ij} \geq 1$$

  De cette manière, nous garantissons d'avoir un support pour l'interprétation de chaque cluster ;
- ***Non-recouvrement*** Un mot ne peut être utilisé dans plusieurs descriptions :

$$\forall j \in [1,m], \sum_{i=1}^{k} X_{ij} \leq 1$$

  Nous évitons ainsi de construire des descriptions trop similaires ;
- ***Taille max*** Une description doit contenir au plus t mots :

$$\forall i \in [1,k], \sum_{j=1}^{m} X_{ij} \leq t$$

  Nous limitons la taille des descriptions pour ne pas compliquer leur interprétation ;
- ***Score non-nul*** Un mot dans une description doit avoir un score DF-IDF non nul dans le cluster correspondant :

$$\forall i \in [1,k], \forall j \in [1,m], P_{ij} = 0 \implies X_{ij} = 0$$

  Du fait des contraintes non-vide et non-recouvrement, le modèle peut affecter un mot à un cluster même s'il n'y apparait pas. Cela n'apportant rien pour l'interpréter, nous prévenons cette possibilité.
- ***Couverture d'apparitions*** La somme du nombre d'apparition des mots d'une description dans le cluster correspondant doit dépasser une proportion α de la taille du cluster :

$$\forall c \in [1,k], \sum_{j=1}^{m}\sum_{i=1}^{|C_c|} T^c_{ij}X_{cj} \geq \alpha|C_c|$$

  Cette contrainte ne représente pas la couverture effective des tweets d'un cluster par les mots de sa description. En effet sur un cluster de deux tweets, si l'un contient deux mots descripteurs et l'autre aucun, une description contenant ces deux mots satisfera cette contrainte avec $\alpha = 1$ même si elle ne couvre effectivement que la moitié de ce cluster. Nous formulons une contrainte de couverture de cette manière pour qu'elle soit la plus riche possible tout en restant linéaire.

4 Expérimentations

Les expérimentations sont conduites pour répondre aux questions suivantes :

- Quelle est la meilleure manière de choisir les descripteurs potentiels que l'on peut affecter à un cluster ?
- Les descriptions construites sont-elles adéquates pour leur clusters ?

Nous apportons nos éléments de réponse à ces deux questions dans la section 4.2.

4.1 Paramétrage expérimental

4.1.1 Données et requête

Pour constituer un premier jeu de données sur lequel évaluer notre approche, nous avons ciblé l'évènement climatique marquant le plus récent en France au moment où nous avons lancé la collecte. C'est donc sur la tempête Alex, qui a frappé le sud-est et le nord-ouest du pays, que nous avons centré notre requête.
Les épisodes climatiques violents sont des cibles toutes indiquées pour vérifier nos hypothèses car ils touchent un territoire et tous les utilisateurs de Twitter qui s'y trouvent. D'autres types d'évènements peuvent en effet n'affecter ou n'intéresser que certaines communautés sur la plate-forme, limitant ainsi la variété du contenu que l'on peut en retirer.

En s'intéressant à une tempête, les messages récoltés seront diversifiés en contenu, forme et origine.

L'idéal pour construire une requête visant à capter des tweets pour caractériser un évènement et ses manifestations serait de disposer d'une ontologie décrivant cette structuration. On disposerait alors de mots clefs pour construire plusieurs requêtes de différentes granularités.
Sans cette ontologie, nous travaillons à partir d'une liste de mots clefs restreinte se résumant dans notre cas à "tempête" et "alex". Nous ajoutons à la requête des variations de ces formes pour capturer les morphologies incorrectes qui peuvent apparaître sur Twitter.

4.1.2 Préparation des données

Les mots vides, ou stop words, et les tokens utilisés dans la requête sont supprimés des tweets collectés. Nous utilisons ensuite le LefffLemmatizer [2] à travers son implantation dans spaCy pour obtenir des lemmes et nous utilisons des expressions régulières pour remplacer les heures et les nombres avec respectivement les étiquettes 'matchedHour' et 'matchedNumber'. Comme le tokeniseur intégré au lemmatiseur s'est montré moins robuste sur nos tweets que sur des jeux de données classiques, nous avons aussi ajouté une expression régulière pour vérifier que la séparation des mots était correcte. Nous n'avons retenu que les lemmes d'au moins trois caractères et les tweets composés d'au moins deux lemmes.

4.1.3 Représentation des données

Pour représenter nos tweets avec des vecteurs, nous utilisons le plongement de document proposé dans [10]. Une fois préparées, nos données sont exploitées par ce modèle pour apprendre un espace de représentation dans lequel la distance entre deux points représente une distance sémantique entre les deux tweets correspondants.
Nous avons choisi le modèle de plongement Doc2Vec car il permet d'apprendre une représentation des documents en même temps qu'une représentation des mots qui les composent. Nous en avons utilisé l'architecture PV-DBoW plutôt que sa contrepartie PV-DM. La première rend l'entraînement plus coûteux que la seconde mais fournit des vecteurs qui mènent à obtenir de meilleurs résultats pour les tâches qui en évaluent la qualité.
Le modèle apprend l'espace de représentation en s'entraînant à prédire le mot venant après une séquence de mots donnés. Dans notre cas cela consiste à extraire une séquence de mots du tweet en cours d'apprentissage, en en conservant l'ordre, pour prédire le dernier mot de cet extrait. La taille de cette séquence est contrôlée. Nous l'avons paramétrée pour que le modèle s'entraîne à prédire le dernier d'une séquence de trois mots. Nous l'avons déterminé empiriquement, une séquence trop longue se révélant inadaptée pour s'entraîner sur des tweets pouvant être plus petits qu'elle.
Nous avons utilisé l'implantation du modèle de Gensim 4.1.2 [3].

4.1.4 Clustering

Nous avons utilisé l'algorithme K-Means pour clusteriser les tweets, en nous appuyant sur l'hypothèse que les tweets avec un sens proche seront proches dans l'espace de plongement. Nous l'avons paramétré avec K=10. Nous avons déterminé ce nombre empiriquement. Trop peu de clusters les font contenir des informations sur plusieurs sujets. Trop de clusters amène à séparer des informations qui devraient être regroupées, comme c'est le cas avec 20 clusters pour nos données. Ce paramétrage est expérimentalement apparu comme un juste compromis pour faire apparaître des clusters bien identifiés et d'autres qui pourraient être soit sous découpés soit réunis. Nous avons ainsi un échantillon de tous les cas qui peuvent nous intéresser. Nous initialisons le clustering avec la méthode K-Means++ [14], une méthode de sélection des centres de cluster initiaux semi-aléatoire qui permet d'optimiser le clustering. Par ailleurs nos expérimentations ont montré que travailler sur des vecteurs de score TF-IDF engendre des clusters déséquilibrés en taille tandis que travailler sur la base de plongements a permis de les harmoniser : avec un corpus de 19457 tweets, 7 clusters font entre 1000 et 3000 tweets, 2 autres moins de 700 et un dernier plus de 4700.

4.1.5 Problème de PLNE

Un framework déclaratif est utilisé pour présenter notre problème. Nous l'avons codé en utilisant l'interface Python 3.8.8 [4] du solver Gurobi 9.1.1.2 [5].

4.1.6 Métriques d'évaluations

À notre connaissance, il n'y a pas de métrique de référence qui serait adaptée à l'évaluation des descriptions que nous construisons. Par exemple, les différentes métriques ROUGE [12] offrent une évaluation en comparant un résumé généré à une référence, dite gold, produite par un humain, qui n'existe pas pour nos données. SUPERT [6] compare les phrases du résumé produit aux phrases saillantes du document. Notre description étant constituée seulement de mots, ce genre de métrique n'est pas approprié non plus. Nous proposons donc les deux critères suivants :

— **Importance Relative (IR).** L'importance relative d'un ensemble de mots d par rapport à un cluster C est définie en utilisant le score DF-IDF :

$$IR(d, C) = \sum_{w \in d} DFIDF(w, C) \qquad (3)$$

Ceci permettra de mesurer la pertinence d'une description par rapport à un cluster.

— **Somme Pondérée des Distances (SDP).** Un autre objectif est de construire des descriptions de clusters sémantiquement cohérentes et distinctes. Nous proposons de mesurer la différence de deux descriptions d_i et d_j par :

$$SPD(d_i, d_j) = \frac{\sum_{w \in d_i} \sum_{w' \in d_j} dist(w, w')}{|d_i||d_j|} \qquad (4)$$

2. https ://spacy.io/universe/project/spacy-lefff
3. https ://radimrehurek.com/gensim/
4. https ://www.python.org/downloads/release/python-388/
5. https ://www.gurobi.com/documentation/9.1

où $dist(w, w')$ est la distance dans l'espace de plongement entre les vecteurs représentant w et w'.

Une IR haute d'une description par rapport à un cluster indiquera que les mots de celles-ci sont adéquats pour identifier le cluster et le séparer des autres, au regard du score DF-IDF. On attend donc une IR d'une description plus élevée pour son cluster que pour les autres. Si une description a une IR similaire pour tous les clusters, elle les identifie et les sépare tous de la même manière et ne permet alors d'en distinguer aucun.
La SPD représente la distance entre deux descriptions. Plus elle est basse, plus elles sont proches dans l'espace de plongement des vecteurs de tweets et ainsi elles sont proches sémantiquement l'une de l'autre. On attend ainsi la SPD d'une description plus basse par rapport à elle-même que par rapport aux autres. Si toutes les SPD d'une description sont équivalentes, alors elle est aussi proche sémantiquement d'elle-même que des autres et il est difficile de déterminer ce qui la caractérise sur ce critère.
Comparer les descriptions avec ces métriques apporte donc des informations sur leur séparation et leur identification, et par extension sur celles des clusters concernés.

4.2 Résultats et analyses

Plusieurs exécutions du clustering avec un même paramétrage basés sur des plongements de documents donnant des clusters très similaires, l'échantillon de résultats présentés est représentatif de ceux que l'on peut obtenir avec notre méthode. Il faut noter que le score obtenu par le modèle d'optimisation construisant les descriptions varie, de manière non significative toutefois : d'un clustering à l'autre, le total des scores des descriptions obtenu varie en moyenne de moins de 0,08%. L'initialisation K-Means++ [14] sélectionne un premier centre de cluster aléatoirement parmi les données puis les centres suivants les plus éloignés les uns des autres pour optimiser le clustering. Nous avons exécuté un grand nombre de fois le clustering sur les mêmes données pour vérifier la robustesse des partitions construites. Les clusterings et les descriptions obtenus sont suffisamment similaires en terme de composition [6] pour être représentatifs de l'ensemble des résultats. Nous en présentons donc un ici, sélectionné aléatoirement.

Nous donnons ici les résultats des sélections de candidats présentés dans la section 3.2.2. Nous fixons le paramètre α de la contrainte de couverture d'apparition présenté dans la section 3.2.3 à 30 pour des raisons de comparaisons. C'est le maximum qu'il est possible de fixer sur cette contrainte commun aux trois méthodes de sélection. Les clusters et leur description sont alignés dans le tableau de résultats. Le meilleur score d'une ligne de matrice apparaît en orange. Ainsi, une sélection de bonne qualité au regard de nos critères correspondra à un tableau avec une diagonale nord-ouest/sud-est fortement colorée avec des valeurs en orange et le reste du tableau faiblement coloré.

6. D'une exécution de clustering à l'autre, la composition des clusters est identique en moyenne à 99,5% et le score total du modèle construisant les descriptions varie en moyenne de 0,08%.

Les expérimentations sont réalisées sur une machine dotée d'un i7-11850H et de 32 Go de RAM. Le temps total d'exécution est inférieur à 2 minutes avec les sélections par les plus hautes fréquences et Tops DF-IDF. Avec la sélection hybride, on rajoute 1 minute 30. À noter que la majorité de ces temps est occupée par les écritures/lectures des fichiers de résultats.

4.2.1 Résultat sélection par les plus hautes fréquences

Id description \ Cluster Id	0	1	2	3	4	5	6	7	8	9
0	3.7625	0.4101	0.6088	0.3916	0.5083	4.0253	0.7407	0.8427	1.0482	0.489
1	0.6384	1.1374	0.4155	0.6609	0.7196	1.5477	0.2794	0.2187	0.2677	0.3943
2	0.0649	0.0663	0.5986	0.0316	0.0809	0.0694	0.5158	0.1034	0.1454	0.0379
3	0.4414	0.9758	0.407	2.6059	1.1519	0.6114	0.5107	0.3891	0.2377	1.7125
4	0.4635	0.8924	0.5764	1.1311	1.1293	0.6666	0.46	0.3805	0.4049	0.9444
5	4.2155	0.6867	0.4911	0.5093	0.7322	3.4251	0.6809	0.4188	0.5048	0.5554
6	1.3401	0.6176	0.9346	0.9989	0.8777	1.3601	1.0043	1.307	1.1801	1.0643
7	1.2352	0.3813	1.051	0.2253	0.6228	1.2396	0.8427	3.3827	3.3522	0.2843
8	0.6688	0.4155	0.6964	0.1727	0.5684	0.641	0.4559	1.8767	4.5704	0.1795
9	0.6819	0.4655	0.3026	1.2289	0.5974	0.7689	0.5159	0.2897	0.1349	1.9666

(a) Matrice de l'importance relative (IR) des descriptions

Id description \ Description Id	0	1	2	3	4	5	6	7	8	9
0	0.2621	0.5695	0.7787	0.6878	0.6612	0.3793	0.5556	0.5246	0.5833	0.5618
1	0.5695	0.3787	0.5076	0.4223	0.4591	0.5009	0.6091	0.6982	0.6719	0.4533
2	0.7787	0.5076	0.1141	0.679	0.6555	0.7046	0.8118	0.74	0.675	0.7642
3	0.6878	0.4223	0.679	0.15	0.3126	0.6864	0.5659	0.7853	0.7921	0.2959
4	0.6612	0.4591	0.6555	0.3126	0.3748	0.6832	0.5897	0.7118	0.6964	0.4092
5	0.3793	0.5009	0.7046	0.6864	0.6832	0.2468	0.6179	0.6612	0.6779	0.583
6	0.5556	0.6091	0.8118	0.5659	0.5897	0.6179	0.468	0.5908	0.6678	0.5354
7	0.5246	0.6982	0.74	0.7853	0.7118	0.6612	0.5908	0.2611	0.4022	0.7262
8	0.5833	0.6719	0.675	0.7921	0.6964	0.6779	0.6678	0.4022	0.1252	0.7657
9	0.5618	0.4533	0.7642	0.2959	0.4092	0.583	0.5354	0.7262	0.7657	0.3053

(b) Matrice des sommes pondérées des distances (SPD) des descriptions

Id	Mots
0	'bretagne', 'violent', 'morbihan', 'météo', 'électricité', 'vendredi', 'fort', 'ouest', 'alerte', 'jeudi'
1	'matchedNumber', 'temps', 'nouveau', 'grand', 'contre', 'entre'
2	'ainsi', 'trois', 'bois', 'bout', 'vivement', 'dansaient', 'capucine', 'virant'
3	'faire', 'bien', 'voir', 'très', 'quand', 'trop', 'dire', 'beau', 'rien', 'journée'
4	'comme', 'fait', 'sans', 'depuis', 'pouvoir', 'aussi', 'venir', 'merci'
5	'vent', '#tempête', 'nuit', 'heure', 'kilomètre', 'rafale', 'vents', '#bretagne', '#morbihan', 'côte'
6	'après', '#tempetealex', 'demain', 'cause', 'arriver', 'dégât', 'image', 'soir', 'vallée'
7	'alpes-maritimes', 'france', 'passage', 'vigilance', 'sinistré', 'personne', 'disparu', 'rouge', 'département', 'crue'
8	'#alpesmaritimes', 'deux', 'direct', 'mort', 'corps', 'pompier', 'bilan', 'italie', 'orange', 'toujours'
9	'aller', 'avant', 'calme', 'chez', 'pluie', 'tempete'

(c) Description des clusters

FIGURE 1 – Évaluation des descriptions obtenues avec la sélection de candidats par les plus hautes fréquences. Cette sélection propose 87 candidats dont 35, apparaissant en vert foncé, ne sont pas dupliqués dans leur liste initiale

Dans la figure 1a nous pouvons voir que même si la diagonale est fortement colorée, la cellule de meilleur score est dans 5 cas sur 10 située ailleurs sur la ligne. Les descriptions produites sont donc, du point de vue leur IR, adéquates pour leur cluster mais également pour d'autres. Les descriptions 0, 1, 4, 6 et 7 sont mêmes plus adéquates pour un autre cluster. On peut remarquer que les descriptions 2, 3, 5 et 8 ont une valeur d'IR pour leur cluster nettement plus élevée. En outre, on remarque que les lignes 1, 4 et 6 et 9 sont fortement colorées. Les descriptions correspondantes ont donc une IR similaire pour tous les clusters.

Dans la figure 1b, la meilleure valeur de la ligne est bien sur la diagonale de la matrice dans 8 cas sur 10. Dans la majorité des cas, les descriptions semblent donc plus proches sémantiquement d'elles-mêmes que d'autres descriptions. Par ailleurs, les lignes fortement colorées sont les 0, 6 et 9. Elles sont relativement peu nombreuses. Il semble donc que les descriptions soient distinctes sémantiquement les unes des autres. On constate des lignes où l'écart entre la meilleure valeur et les autres est net (1, 2, 3, 5 et 8) et d'autres plus

uniformes (0, 6 et 9). Les observations sur ces deux matrices semblent pointer vers les conclusions suivantes : soit il y a des clusters particulièrement faciles et d'autres particulièrement difficiles à distinguer, soit la sélection par fréquence n'offre pas de candidats adaptés pour pouvoir décrire les clusters de manière distincte.

Dans la figure 1c, on peut constater que moins de la moitié des mots à la disposition du modèle de PLNE n'étaient pas dupliqués dans la liste initiale des candidats par la fréquence. Il est possible d'interpréter cela de deux manières différentes : soit le clustering peine à séparer certains clusters, soit la fréquence seule n'est un pas un critère suffisant pour identifier des descripteurs distincts d'un cluster à l'autre. Du point de vue de l'interprétation, on remarque que certaines descriptions sont beaucoup plus compréhensibles que d'autres. Par exemple on voit clairement se dégager un sujet pour le cluster 5, les vents qui ont frappé les côtes de la Bretagne lors d'une nuit de la tempête, là où il est impossible d'attribuer un sujet en particulier à la description 3. Les descriptions 0, 5, 7 et 8 paraissent ainsi assez facilement interprétables, les autres ne semblant centrées sur rien en particulier.

4.2.2 Résultat sélection par les Tops DF-IDF

Id description \ Cluster Id	0	1	2	3	4	5	6	7	8	9
0	3.7625	0.4101	0.6088	0.3916	0.5083	1.0218	0.7407	0.8427	1.0482	0.489
1	0.3886	1.4063	0.4958	0.747	0.7511	0.7779	0.3132	0.3642	0.4399	0.4258
2	0.0917	0.2096	1.0940	0.1003	0.2291	0.0747	0.6342	0.4301	0.5152	0.1029
3	0.2981	0.8142	0.3661	2.3409	1.0559	0.3939	0.4806	0.2755	0.1476	1.511
4	0.4218	1.131	0.7816	1.3004	1.3309	0.5638	0.512	0.6415	0.7335	0.8475
5	4.407	0.946	0.5667	0.5719	0.8897	2.0439	0.7195	0.4374	0.518	0.5943
6	1.8286	0.6169	0.9092	0.9042	0.8464	2.105	1.0216	1.4498	1.4411	0.9264
7	1.1049	0.2756	0.7668	0.1813	0.5214	1.2222	0.6732	2.7907	3.13	0.2335
8	0.5277	0.3701	0.7693	0.111	0.5899	0.4122	0.5313	2.4536	2.1217	0.1241
9	0.8505	0.7084	0.4292	2.0592	0.9755	0.9627	0.7647	0.4009	0.1882	2.2341

(a) Matrice de l'importance relative (IR) des descriptions

Id description \ Description Id	0	1	2	3	4	5	6	7	8	9
0	0.7625	0.5336	0.7878	0.7132	0.6587	0.3985	0.4723	0.5402	0.6089	0.5943
1	0.5336	0.3847	0.5093	0.4456	0.4978	0.5251	0.5425	0.5655	0.5462	0.4784
2	0.7878	0.5093	0.7625	0.6383	0.6427	0.7327	0.7742	0.6725	0.6126	0.749
3	0.7132	0.4456	0.6383	0.4577	0.3934	0.6983	0.6435	0.7786	0.8069	0.2587
4	0.6587	0.4978	0.6427	0.3934	0.4508	0.6885	0.5929	0.6383	0.655	0.4558
5	0.3985	0.5251	0.7327	0.6983	0.6885	0.4577	0.5377	0.6924	0.7046	0.6087
6	0.4723	0.5425	0.7742	0.6435	0.5929	0.5377	0.4806	0.5888	0.642	0.5739
7	0.5402	0.5655	0.6725	0.7786	0.6383	0.6924	0.5888	0.3795	0.3617	0.7512
8	0.6089	0.5462	0.6126	0.8069	0.655	0.7046	0.642	0.3617	0.4154	0.8175
9	0.5943	0.4784	0.749	0.2587	0.4558	0.6087	0.5739	0.7512	0.8175	0.2808

(b) Matrice des sommes pondérées des distances (SPD) des descriptions

Id	Mots
0	'vent', 'morbihan', 'violent', 'électricité', 'météo', 'ouest', 'fort', 'alerte', 'vendredi', 'jeudi'
1	'matchedNumber', 'temps', 'nouveau', 'grand', 'contre', '#france', 'entre', '#lemonde', 'place', 'suite'
2	'ainsi', 'bois', 'dansaient', 'capucine', 'virant', 'bout', 'vivement', 'trois', 'sinistré', 'village'
3	'voir', 'bien', 'quand', 'dire', 'beau', 'vouloir', 'rien', 'journée', 'prendre', 'bonjour'
4	'faire', 'comme', 'fait', 'sans', 'pouvoir', 'aussi', 'venir', 'jamais', 'vallée', '#alex06'
5	'#tempête', 'nuit', 'heure', 'kilomètre', 'rafale', 'vents', '#bretagne', '#morbihan', 'côte', 'terre'
6	'#tempetealex', 'après', 'aller', 'bretagne', 'arriver', 'dégât', 'france', 'image', 'soir', 'premier'
7	'alpes-maritimes', 'vigilance', 'passage', 'personne', 'disparu', 'rouge', 'crue', 'département', 'recherchées', 'habitant'
8	'#alpesmaritimes', 'deux', 'direct', 'mort', 'corps', 'pompier', 'bilan', 'italie', 'retrouvé', 'applegreen'
9	'depuis', 'avant', 'calme', 'demain', 'chez', 'trop', 'cause', 'pluie', 'tempete', 'dehors'

(c) Descriptions des clusters

FIGURE 2 – Évaluation des descriptions pour la sélection de candidat par les Top DF-IDF, avec 101 candidats dont 51 non dupliqués.

Dans la figure 2a nous pouvons voir la diagonale fortement colorée et la cellule de meilleur score d'une ligne dans 7 cas sur 10 sur la diagonale. Les descriptions produites sont donc, du point de vue leur IR, adéquates pour leur cluster et la majorité d'entre elles sont les plus adaptées pour leur cluster. On peut remarquer que les descriptions 2, 3, 5 et 8 ont une valeur d'IR pour leur cluster qui se distingue nettement des autres. Dans 4 cas sur 10 donc, la description discrimine nettement mieux son cluster que les autres. On remarque d'autre part que les lignes 1, 4, 6, et 9 sont fortement colorées. Les descriptions sont donc dans 4 cas sur 10 similaires aux autres du point vue IR.

Dans la figure 2b, la meilleure valeur de la ligne est sur la diagonale dans 6 cas sur 10. On peut remarquer que pour les descriptions 6, 7 et 9, la valeur de la diagonale est tout de même proche de la meilleure valeur de la ligne. Il semble donc que la sélection par DF-IDF propose des candidats qui permettent de construire des descriptions cohérentes mais qui peinent à se distinguer sémantiquement. Par ailleurs, on peut remarquer que les lignes 0, 6 et 9 sont fortement colorées. On constate ici des tendances similaires avec les résultats de la sélection par les plus hautes fréquences. Nous pouvons l'interpréter de deux manières différentes : soit la composante fréquentielle du calcul du score DF-IDF y est encore trop forte et cette sélection pose le même problème que la précédente, soit il y a des clusters plus difficiles à discerner des autres.

Dans la figure 2c, on constate que plus de la moitié des candidats initiaux n'étaient pas dupliqués. Les sujets des descriptions des clusters 0, 5, 6, 7 et 8 semblent se démarquer assez facilement alors que ceux des autres sont beaucoup moins clairs voire impossibles à identifier.

4.2.3 Résultat sélection hybride

Id description \ Cluster Id	0	1	2	3	4	5	6	7	8	9
0	2.2925	0.3458	0.3741	0.3243	0.3897	0.4316	0.3999	0.4423	0.6993	0.3419
1	0.5901	1.3084	0.539	0.7565	0.8569	1.1598	0.3357	0.5115	0.6812	0.4923
2	0.2562	0.6439	1.4282	0.3565	0.6358	0.34	0.5215	1.0348	1.0925	0.2499
3	0.3958	0.6206	0.4111	1.4974	0.6343	0.5433	0.4317	0.2432	0.2422	0.7947
4	0.5731	0.9287	0.5976	1.1377	1.1423	0.7669	0.4401	0.4581	0.4584	0.7431
5	3.2166	0.9414	0.5286	0.549	0.7976	0.3663	0.5644	0.304	0.4037	0.5044
6	1.328	0.5612	0.8625	0.934	0.8703	1.0826	1.009	1.3284	1.207	1.1123
7	0.3262	0.1965	0.4959	0.2471	0.4124	0.4267	0.3843	1.434	1.3691	0.2623
8	0.484	0.4197	0.7192	0.2389	0.5704	0.3861	0.3665	1.4489	1.2367	0.2293
9	0.303	0.5787	0.2732	1.3423	0.6218	0.2186	0.4974	0.2837	0.103	1.2351

(a) Matrice de l'importance relative (IR) des descriptions

Id description \ Description Id	0	1	2	3	4	5	6	7	8	9
0	0.2925	0.4984	0.7308	0.6072	0.5431	0.3721	0.5503	0.6004	0.6369	0.5848
1	0.4984	0.3426	0.6068	0.4831	0.4817	0.5161	0.5092	0.519	0.5753	0.4683
2	0.7308	0.6068	0.4282	0.6409	0.6238	0.7639	0.639	0.4677	0.5284	0.6542
3	0.6072	0.4831	0.6409	0.3048	0.4197	0.6225	0.4949	0.5798	0.692	0.5952
4	0.5431	0.4817	0.6238	0.4197	0.439	0.5585	0.5277	0.5483	0.6209	0.6032
5	0.3721	0.5161	0.7639	0.6225	0.5585	0.2925	0.6011	0.6822	0.7201	0.6021
6	0.5503	0.5092	0.639	0.4949	0.5277	0.6011	0.4316	0.5411	0.6512	0.499
7	0.6004	0.519	0.4677	0.5798	0.5483	0.6822	0.5411	0.3284	0.3847	0.5782
8	0.6369	0.5753	0.5284	0.692	0.6209	0.7201	0.6512	0.3847	0.3691	0.6783
9	0.5848	0.4683	0.6542	0.2841	0.3652	0.6021	0.499	0.5782	0.6783	0.4284

(b) Matrice des sommes pondérées des distances (SPD) des descriptions

Id	Mots
0	'électricité', '#tempêtes', 'pluie', 'tropical', 'dimanche', 'violent', 'alerte', 'inondation', 'jusqu'', 'delta'
1	'temps', 'autre', 'nouveau', 'grand', 'place', 'rester', 'octobre', 'samedi', '#france', 'matchedHour'
2	'depuis', 'sinistré', '#tempêtealex', 'solidarité', 'parc', 'pont', 'trois', 'vallée', 'soutien', 'virant'
3	'apres', 'falloir', 'hier', 'voir', 'très', 'alors', 'voici', 'parler', 'parce', 'plage'
4	'sans', 'jamais', 'faire', 'cette', 'jours', 'avis', 'aide', 'aussi', 'raison', 'face'
5	'vent', 'matchedNumber', 'côte', 'ouragan', 'heure', 'dépression', 'vers', 'entre', 'kilomètre', '#morbihan'
6	'demain', 'avant', 'image', 'cause', 'france', 'après', 'fermé', 'dégât', 'premier', 'jour'
7	'point', 'secours', 'nice', 'annoncer', 'crue', 'maison', 'région', 'disparu', 'emporté', 'épisode'
8	'emmanuel', 'macron', 'corps', 'rouge', 'moins', '#nice06', 'disparus', 'plusieurs', 'toujours', 'village'
9	'seul', 'comment', 'gros', 'calme', 'chez', 'fois', 'quoi', 'rien', 'ciel', 'dire'

(c) Descriptions des clusters

FIGURE 3 – Évaluation des descriptions pour la sélection de candidat hybride, avec 118 candidats dont 72 non dupliqués.

L'utilisation de l'algorithme FPF pour orienter la sélection des candidats permet de mieux couvrir la sémantique représentée par les vecteurs des mots des tweets d'un cluster. Ce gain de représentativité s'accompagne d'un autre effet. En choisissant des candidats à la périphérie de la représentation sémantique d'un cluster, on les rapproche naturellement des autres clusters même si cet effet est compensé par l'imposition d'un seuil de DF-IDF.

Nous pouvons le voir notamment dans la figure 3a. La meilleure valeur d'une ligne est sur la diagonale dans 7 cas sur 10. La matrice apparaît fortement colorée. Les lignes 1, 2, 4, 6, et 7 apparaissent presque uniformes. Au regard de ce critère, la sélection hybride semble proposer des candidats adéquats pour décrire les clusters mais pas pour différencier les descriptions. Nous pouvons remarquer toutefois que la valeur des diagonales des lignes 3, 5 et 8 se distinguent nettement des autres, ce qui semble indiquer que certains clusters sont plus faciles à différencier dans leur description. De manière générale les scores d'IR sont plus bas que pour les autres sélection, ce qui s'explique aussi par l'utilisation de FPF dans la sélection des candidats.

Cette tendance s'observe aussi dans la figure 3b. La meilleure valeur de la ligne se trouve sur la diagonale dans 8 cas sur 10 mais n'est nettement meilleure que dans 2 cas sur 8 (lignes 5 et 9). Comme pour l'IR, la SPD les descriptions obtenues avec la sélection hybride pourraient indiquer que les candidats de cette sélection sont adéquats pour décrire les clusters mais pas pour les isoler.

Dans la figure 3c, on peut voir qu'un peu plus d'un tiers des descripteurs initiaux de la sélection hybride sont dupliqués. C'est significativement moins qu'avec les autres méthodes. Le sujet des descriptions 2, 5, 6, 7 et 8 semble s'identifier assez facilement.

4.2.4 Comparaison de l'ensemble des méthodes

Nous comparons ici les trois approches présentées sur l'IR, la SPD et la couverture des données.

La figure 4 permet de voir qu'elles ont des performances différentes en terme d'IR mais similaires en terme de SPD. Nous voyons également apparaître que la sélection basée uniquement sur la fréquence est moins bonne que les deux autres au regard de ces critères. La sélection hybride et celle des Tops DF-IDF affichent des résultats similaires, l'hybride étant légèrement meilleure que les Tops DF-IDF.

En observant ces résultats, il apparaît que la fréquence seule ne suffit pas à identifier des descripteurs qui permettent de construire des descriptions appropriées pour leur cluster. Lorsqu'on s'appuie sur ce seul critère, on peut en effet sélectionner des mots très fréquents dans un cluster avec un score DF-IDF de fait plus bas que dans d'autres clusters. Ainsi, ce descripteur sera plutôt utilisé dans un autre cluster où son score sera plus élevé. De cette manière, l'ensemble des candidats présentés au modèle d'optimisation peut dans un cas non favorable compter une minorité de mots importants pour décrire les clusters. Les descriptions qui sont alors construites seront peu adéquates pour leur ensemble de tweets.

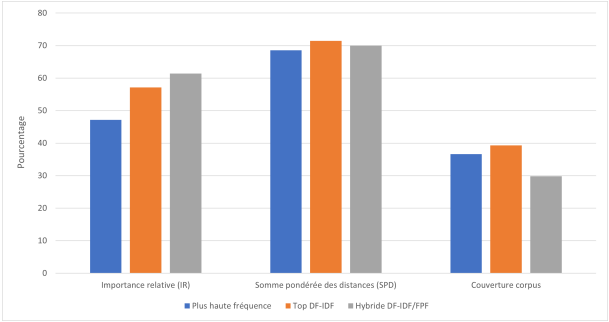

FIGURE 4 – Proportion des cas où le meilleur score d'une description est le meilleur pour le cluster qu'elle décrit et couverture du corpus par les descriptions

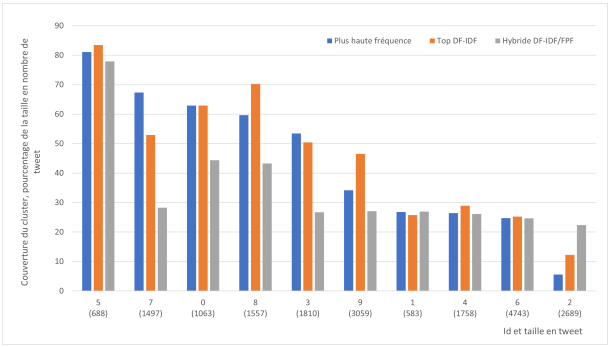

FIGURE 5 – Couverture des clusters par sélection de candidat

Nous pouvons voir, dans la figure 4 la couverture du corpus selon la sélection de candidat. Si la sélection par plus haute fréquence et des Tops DF-IDF ont des résultats similaires, respectivement 36,6% et 39,3%, la sélection hybride se montre moins performante au regard de critère, avec 29,9% de tweets du corpus couverts. En s'appuyant, même en partie, sur l'algorithme Farthest Point First pour identifier des candidats pour chaque cluster, on sélectionne des mots éloignés les uns des autres dans l'espace de plongement. Par construction les mots plus proches apparaissent souvent ensemble. De ce fait, les candidats sélectionnés par l'approche hybride apparaissent dans moins de tweets et la couverture du corpus par les descriptions obtenues en pâtit.

Nous pouvons constater la même tendance, s'expliquant de la même manière, dans les couvertures par cluster de la figure 5. Nous peuvons aussi y constater que même si de manière générale l'approche hybride sélectionne des descripteurs moins couvrants, elles est globalement plus stable et s'en sort au moins aussi bien sinon mieux avec les clusters difficiles à couvrir, comme les 4, 6 et 2.

La couverture représentée dans les figures 4 et 5 n'est pas la même que celle imposée par les contraintes du modèle, voir section 3.2.3. Ici c'est la couverture réelle des descriptions qui est présentée, soit la proportion de tweets du corpus ou du cluster couverte. Ces deux mesures peuvent concorder, pour la couverture de l'ensemble du corpus par exemple (figure 4), mais aussi être complètement déconnectées. Nous pouvons le voir dans la figure 5 pour le cluster 2, dont seulement 5,6% des tweets sont couverts alors que c'est le cas pour 36,3% des apparitions des candidats dans ce cluster et

que la contrainte est respectée.

Les résultats des différentes sélections montrent des tendances communes. D'abord, le manque de variété au sein des candidats sélectionnés induit de moins bons résultats. Par ailleurs si une description peut poser problème à une approche et pas aux autres, certains clusters semblent de manière générale plus faciles ou plus difficiles à décrire. Par exemple, le cluster 5 est systématiquement bien identifié, tant en IR qu'en SPD. À l'inverse, le 4 pose toujours problème. Il apparaît donc qu'indépendamment des descripteurs potentiels choisis, certains clusters soient moins favorables à la construction de descriptions avec de bonnes IR et SPD. Le même problème s'observe avec la couverture, avec des clusters plus difficiles à couvrir que d'autres comme on peut le voir sur la figure 5.
Construire une description pour un cluster de tweets avec notre méthode comprend donc deux axes de difficulté : représenter ce sur quoi porte le contenu du cluster pour l'interpréter et couvrir ses tweets dans une proportion significative pour que l'interprétation soit robuste. Les résultats en IR et SPD permettent de se faire une idée de la difficulté sur le premier axe, la mesure de la couverture d'un cluster par sa description permettant de le faire pour l'autre. Intuitivement on pourrait se dire que ces deux difficultés sont liées, qu'il faut couvrir un cluster pour en capturer le sens. Les évaluations des clusters 5 et 6 semblent pointer dans cette direction. Pourtant le cluster 2, avec de bonnes IR et SPD apparaît particulièrement difficile à couvrir. Il n'est donc pas possible d'établir l'existence ou l'absence d'un lien entre ces deux axes.

Plusieurs questions se posent sur l'interprétabilité humaine des clusters et de leurs descriptions. Par exemple, les scores d'IR et de SPD de la description du cluster 5 sont toujours bons et quelle que soit la méthode. On y retrouve toujours certains mots : 'vent', '#morbihan', 'côte', 'kilomère' et 'heure'. Dans les sélections par les plus hautes fréquences et Tops DF-IDF, on y trouve 'rafale'. Dans la sélection hybride, 'rafale' n'est pas présent mais 'ouragan' apparaît. À travers la construction des clusters, l'hypothèse distributionnelle semble se vérifier et permettre d'interpréter facilement ce genre de cluster. Les vecteurs de tweets contenant 'vent' et '#morbihan' semblent proches dans l'espace de plongement, suffisamment pour que ces mots soient systématiquement choisis pour décrire leur cluster.
En observant les descriptions des clusters qui sont consistantes en scores et en contenu avec les différentes sélections, nous pouvons construire une image de la tempête Alex. Au moins une des nuits lors de la tempête a été marquée par des vents très violents sur les côtés bretonnes dans le Morbihan (cluster 5). Par ailleurs les crues dans les alpes maritimes, en particulier dans la région de Nice, ont causé des dégâts et provoqué des disparition à l'origine d'une mobilisation notable des pompiers (clusters 7 et 8).

Pour le cluster 4 en revanche, les choses sont moins claires. Ses trois descriptions présentées ont des scores indiquant qu'elles pourraient convenir à d'autres clusters et aucun objet particulier ne s'en dégage. Elles contiennent en majorité des mots outils : 'fait', 'après' ou encore 'comme'. Outre les pistes d'explications déjà évoquées, cela pourrait venir du fait que certains descripteurs portant une sémantique moins claire et présentant des ambiguïtés (l'ambiguïté n'étant par ailleurs pas traitée dans l'espace de plongement) ne peuvent décrire un sous-évènement en particulier, tel qu'on le constate dans le cluster 5. En effet, les mots outils peuvent apparaître dans n'importe quel contexte, indépendamment du sujet abordé. Par construction, les vecteurs qui contiennent plusieurs de ces mots sont donc naturellement plus éloignés des zones de l'espace sémantique correspondant à un sujet particulier.

5 Conclusion et perspectives

Nous présentons ici deux méthodes de sélection de mots depuis des ensembles de tweets pour décrire au mieux ces ensembles, par les Tops DF-IDF et par la méthode hybride. Si la validation de ces résultats par l'expérimentation sur d'autres jeux de données et l'utilisation d'autres métriques d'évaluation sont nécessaires pour les confirmer, nos premières conclusions sont prometteuses. En effet, dans les cas où les clusters semblent concerner un sous-évènement en particulier qui se distingue du reste des données, nos descriptions permettent de le représenter avec les mots sélectionnés. Ces derniers sont sémantiquement cohérents, entre eux et par rapport au cluster décrit. Nous envisageons plusieurs pistes d'amélioration. Tout d'abord l'utilisation de n-grammes plutôt que d'uni-grammes pour les composants des descriptions, notamment pour se défaire de l'ambiguïté de mots comme "faire". Nous souhaitons également mettre au point des contraintes sur les distances entre les mots et les documents dans l'espace de plongement sémantique. Leur élaboration devra passer par une étude poussée des liens entres les vecteurs de mots et de documents dans cet espace. Par ailleurs, nous souhaitons nous appuyer sur les descriptions pour identifier les clusters qui pourraient être re-découpés et ceux qui pourraient être fusionnés au regard de leur interprétation. C'est un axe qui devra passer par l'étude dans le détail de l'espace de plongement utilisé notamment pour comprendre comment identifier un cluster impossible à interpréter, comme un cluster de mot outil, d'un autre.

Références

[1] J. Ansah, L. Liu, W. Kang, J. Liu, and J. Li. Leveraging burst in twitter network communities for event detection. *World Wide Web*, 2020.

[2] H. Becker, M. Naaman, and L. Gravano. Beyond Trending Topics : Real-World Event Identification on Twitter. *ICWSM*, 2011.

[3] T. Dao, C. Kuo, S. S. Ravi, C. Vrain, and I. Davidson. Descriptive clustering : ILP and CP formulations with applications. In *IJCAI 2018*, pages 1263–1269, 2018.

[4] I. Davidson, A. Gourru, and S. Ravi. The Cluster Description Problem - Complexity Results, Formulations and Approximations. *NeurIPS*, 2018.

[5] De Boom, C. and Van Canneyt, S. and Dhoedt, B. Semantics-driven event clustering in Twitter feeds. In *Proceedings of the 5th Workshop on Making Sense of Microposts*, 2015.

[6] Y. Gao, W. Zhao, and S. Eger. SUPERT : Towards New Frontiers in Unsupervised Evaluation Metrics for Multi-Document Summarization. *ACL*, 2020.

[7] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 1985.

[8] Z. S. Harris. Distributional Structure. *WORD*, 1954.

[9] M. Hasan, M. A. Orgun, and R. Schwitter. TwitterNews+ : A Framework for Real Time Event Detection from the Twitter Data Stream. In *Social Informatics*, Cham, 2016.

[10] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. *CoRR*, 2014.

[11] C. Li, A. Sun, and A. Datta. Twevent : Segment-based event detection from tweets. In *CIKM*, 2012.

[12] C.-Y. Lin and F. Och. Looking for a few good metrics : Rouge and its evaluation. In *Ntcir workshop*, 2004.

[13] A. Ouali, S. Loudni, Y. Lebbah, P. Boizumault, A. Zimmermann, and L. Loukil. Efficiently Finding Conceptual Clustering Models with Integer Linear Programming. In *IJCAI*, 2016.

[14] E. Sherkat, J. Velcin, and E. E. Milios. Fast and Simple Deterministic Seeding of KMeans for Text Document Clustering. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. 2018.

[15] T. Tang and G. Hu. Detecting and Tracking Significant Events for Individuals on Twitter by Monitoring the Evolution of Twitter Followership Networks. *Information*, 2020.

[16] B. C. Wallace, S. Saha, F. Soboczenski, and I. J. Marshall. Generating (Factual ?) Narrative Summaries of RCTs : Experiments with Neural Multi-Document Summarization. *AMIA Summits on Translational Science Proceedings*, 2021.

[17] J. Weng and B.-S. Lee. Event detection in twitter. In *ICWSM*, 2011.

[18] Y. You, G. Huang, J. Cao, E. Chen, J. He, Y. Zhang, and L. Hu. GEAM : A General and Event-Related Aspects Model for Twitter Event Detection. In *WISE*, 2013.

[19] D. Zhou, L. Chen, and Y. He. An unsupervised framework of exploring events on twitter : Filtering, extraction and categorization. In *AAAI*, 2015.