Représentation Temporelle des Mots : Application au Clustering de Micro-Blogs

Želko Kraljevic* — **Nicolas Baskiotis** * — **Benjamin Piwowarski** * — **Patrick Gallinari***

* *Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu 75005 Paris. Email: prenom.nom@lip6.fr*

RÉSUMÉ. *Les modèles distribués de mots sont un outil précieux pour la classification, le clustering, et plus généralement la représentation des documents. Pour des collections dynamiques, il est nécessaire de prendre en compte l'évolution temporelle de ces représentations. Dans cet article, nous présentons un modèle qui représente les mots sous la forme de trajectoires dans un espace de représentation, trajectoires qui sont déterminées par les groupes auxquels les mots appartiennent. Des expériences préliminaires en clustering sur des micro-blogs montrent l'intérêt de ce type de modèle.*

ABSTRACT. *Distributed term models are a powerful tool for classifying, clustering and representing documents. For dynamic collections, we need to model both temporal and topical evolution. In this work, we present a model that uses a continuous time distribution and that represent words as trajectories in a continuous space. We perform some preliminary experiments on Twitter data, showing the potential of our model.*

MOTS-CLÉS : *représentation distribuée, temps, clustering, micro-blogs.*

KEYWORDS: *distributed representation, time, clustering, micro-blogs.*

1. Introduction

La représentation distribuée (ou latente) des mots est un outil précieux pour explorer et analyser les collections de documents textuels. Pour des collections dynamiques, il faut prendre en compte à la fois les dimensions thématiques et temporelles. Ceci permet par exemple d'étudier l'évolution des idées dans la littérature scientifique (Hall *et al.*, 2008) ou des controverses (Beykikhoshk *et al.*, 2015). La plupart des modèles existants reposent sur une discrétisation du temps, ce qui limite leur utilité : ces modèles ne peuvent pas s'adapter de manière fine aux données car le choix de la granularité (heure, jour, etc.) a un impact important sur les solutions trouvées. Une granularité trop large (ex. un mois) ne permettra pas de détecter des événements courts, alors qu'une granularité trop fine (ex. une heure) peut à la fois introduire du bruit et nécessiter de plus grandes ressources (mémoire et temps de calcul).

Dans cet article, nous proposons un modèle dynamique de représentation des mots qui prédit la trajectoire de la représentation de mots dans un espace vectoriel thématique, et où les groupes (clusters) sont définis par une distribution sur le temps, ce qui définit leur influence temporelle, ainsi que par une transformation dans l'espace de représentation. Dans cet article, nous considérons deux transformations qui sont toutes deux des translations. Les transformations ainsi définies sont continues, i.e. la position d'un mot dans l'espace thématique est une fonction continue du temps.

(Mikolov *et al.*, 2013) a proposé un critère qui permet d'apprendre de manière non supervisée la représentation de mots dans un espace vectoriel. Dans cet article, nous proposons d'étendre ce critère pour intégrer le temps dans le contexte. Il est ainsi possible d'apprendre de manière non supervisée les paramètres de chaque groupe (pour le temps et la transformation), ainsi que les degrés d'appartenance des mots à ces groupes. Dans cet article, nous utilisons une collection de micro-blogs qui nous permet d'évaluer les qualités des représentations obtenues.

Dans la suite, nous présentons tout d'abord les travaux liés (section 2), avant de présenter le principe de l'apprentissage des représentations "Word2Vec" (section 3). Puis nous présentons notre modèle de représentation temporel (section 4) avant de présenter des résultats expérimentaux en clustering (section 5).

2. Travaux connexes

La représentation distribuée des mots a une longue histoire dans les domaines de l'accès à l'information et du traitement automatique du langage naturel. Des méthodes de factorisation matricielle (SVD ou LSA) ont été proposées dans (Deerwester *et al.*, 1990) pour trouver des thèmes latents dans une collection de documents. Plus récemment, porté par les succès des approches basées sur des réseaux de neurones pour les modèles de langues, (Mikolov *et al.*, 2013) ont proposé un modèle probabiliste simple, où la probabilité qu'un mot apparaisse à proximité d'un autre dépend directement du produit scalaire entre les représentations vectorielles des deux mots.

Il a été démontré empiriquement que ce modèle, en plus d'être rapide à apprendre, permettait de capturer de façon implicite des relations linéaires entre les mots.

Aucun de ces modèles ne prend par contre en compte le temps, ce qui est une limite pour étudier des collections qui évoluent avec le temps, comme les informations, les (micro-)blogs, ou la littérature scientifique. La plupart des extensions proposées supposent un temps discret en utilisant des fenêtres glissantes sur le temps (qui peuvent se chevaucher ou former une partition). (Saha et Sindhwani, 2012) utilisent une factorisation matricielle non négative pour chaque temps t avec une contrainte de régularité qui force les thèmes entre les temps t et $t + 1$ à être proches. (Wang *et al.*, 2012) ont proposé de prédire l'évolution d'une distribution de probabilité sur les mots en utilisant une transformation linéaire pour modéliser les changements entre deux microblogs d'un utilisateur donné. Contrairement à ce type d'approches, le travail proposé ici ne suppose pas un temps discret, et est plus proche du travail de (Wang et McCallum, 2006). Ceux-ci utilisent une modification du modèle LDA (Blei *et al.*, 2003), où chaque thème est défini par une distribution de probabilité sur le temps ainsi que sur les mots. Dans notre travail, les groupes sont aussi définis en fonction du temps, mais contrairement à (Wang *et al.*, 2012), nous sommes intéressés par la représentation d'un mot à un temps t donné.

Finalement, nos travaux sont aussi liés au domaine de la détection de thèmes (Liu, 2009) qui a pour but de détecter des nouvelles informations et à les suivre. Les modèles utilisés pour cette tâche sont généralement basés sur un clustering incrémental. Par exemple, dans le cadre des micro-blogs, (Rosa *et al.*, 2011) regroupent les microblogs en utilisant l'algorithme des k-moyennes et LDA. Un exemple représentatif du regroupement incrémental est (Chen *et al.*, 2013) où à chaque pas de temps, un microblog peut être classifié dans un groupe pré-existant, ou un nouveau groupe peut être créé. Dans ce travail, nous évaluons notre modèle sur un corpus de micro-blogs, et nous le comparons au travail de (Rosa *et al.*, 2011).

3. Le modèle Word2Vec

Un des récents travaux à succès est le modèle *Word2Vec* (Mikolov *et al.*, 2013) qui propose une architecture de réseau de neurones simple permettant de prédire un terme dans une fenêtre contextuelle prédéfinie. Ce modèle est rapide à entraîner, et permet de calculer des représentations de mots sur de grandes collections. Il a été montré expérimentalement que cette représentation encode implicitement les relations syntaxiques et sémantiques entre les mots : par exemple, il existe une translation dans l'espace des représentations qui transforme la représentation d'un mot singulier (par exemple « ordinateur ») en sa représentation pluriel (« ordinateurs »).

Plusieurs modèles sont proposés dans (Mikolov *et al.*, 2013) ; dans ce travail, nous nous basons sur le modèle nommé *skip-gram* dont le fonctionnement est illustré Figure 1. Le modèle skip-gram a pour but, pour n'importe lequel des mots des documents, de prédire les mots apparaissant dans un voisinage proche de ceux-ci. Dans l'exemple

de la figure 1, le modèle skip-gram utilise la représentation du mot 1 pour prédire la probabilité d'occurrence des mots 2 ou 3.

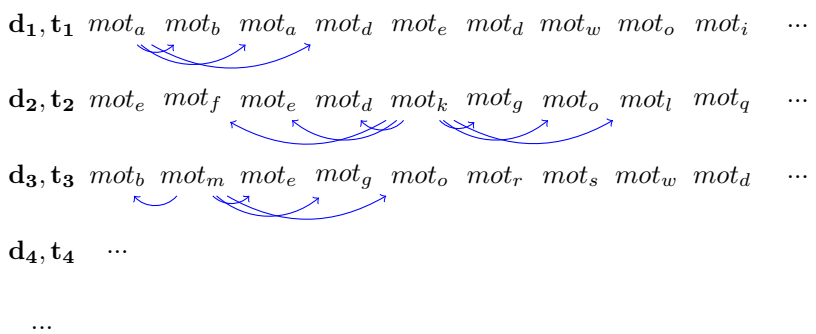

Figure 1. *Documents observés dans une collection de documents. Chaque document correspond à une ligne, et les premiers mots du documents sont donnés pour les trois premiers : le document d_1 commence par le mots a, b, a, d...*

La fonction objectif définie dans word2vec (Mikolov *et al.*, 2013) est la vraisemblance d'observer des mots sachant leur contexte :

$$\mathcal{L} = \prod_{(a,b)\in\mathcal{D}} p(b \mid a)^{\#a} \quad [1]$$

où $\mathcal{D}$ est l'ensemble des observations, ou plus précisément l'ensemble des couples (a, b) tels que le mot a apparaisse près de b (à une distance inférieur à un seuil fixé à 5 comme dans (Mikolov *et al.*, 2013)), et $\#a$ est le nombre de fois où a apparait dans la collection. Dans l'exemple de la figure 1, les couples (a,b), (a, a), (k, f) font partie de $\mathcal{D}$.

Le lien avec la représentation des mots est donné par la façon avec laquelle est calculée la probabilité $p(b \mid a)$, à savoir

$$p(b|a) = \frac{1}{K_a} \exp(x_a \cdot y_b) \quad [2]$$

où x_a et y_b sont les représentations des mots, et K_a une constante de normalisation. Notons qu'il y a deux représentations : une pour le contexte et l'autre pour la cible.

En pratique, une approximation de cette fonction objectif est utilisée, car calculer la normalisation K_a est trop coûteux en temps. Dans cet article, nous suivons la méthodologie de (Mikolov *et al.*, 2013) en utilisant un échantillonnage négatif (NEG)

$$\mathcal{C} = \sum_{(a,b)\in\mathcal{D}} (\log \sigma(x_a \cdot y_b) + k\mathbb{E}\left[\log \sigma(-x_a \cdot y'_b)\right]) \quad [3]$$

où $\sigma(x) = 1/(1 + \exp(-x))$ est la fonction sigmoïde et l'espérance est estimée sur un échantillon de taille [1] k. Cette fonction objectif remplit le même rôle que le maximum de vraisemblance : elle rapproche les représentations x_a et y_b des mots qui co-occurrent, et éloigne les représentations x_a et $y_{b'}$ des mots qui ne co-occurrent pas.

4. Représentation Distribuée Temporelle des Mots

Une première méthode pour étendre le modèle (Mikolov *et al.*, 2013) serait d'utiliser un temps discret, et d'avoir une représentation de chaque mot pour chaque temps donné. Outre le problème du choix de la granularité de temps, cette solution est coûteuse en espace nécessaire pour stocker la représentation de tous les mots.

La solution que nous proposons permet d'éviter ces deux problèmes, en utilisant un temps continu et des groupes de mots (non exclusifs). La classe de modèles que nous proposons est basée sur l'intuition suivante : les mots sont représentés dans un espace thématique, comme pour dans les autres modèles distribués. La différence principale est que cette "position" n'est pas fixe mais dépend du temps.

Plus précisément, les mots ont des trajectoires dans cet espace thématique ; ces trajectoires sont définies par un ensemble de groupes $\mathcal{C}$ appris comme décrit plus loin (section 4.3). Chaque groupe $c \in \mathcal{C}$ est associé avec des paramètres liés au temps, qui définissent *avec quelle magnitude* la position d'un mot est modifiée à un temps t donné, ainsi qu'à des paramètres liés à l'espace thématique, qui défissent *comment* la position d'un mot est modifiée. Dans la suite, nous définissons deux modèles de transformation différents.

Formellement, la représentation $x_a^{(t)}$ d'un mot a au temps t est donnée par :

$$x_a^{(t)} = x_a + \sum_{c \in \mathcal{C}} \rho_{ac} f_c(t) g_c(x_a) \qquad [4]$$

Cette expression est composée de deux termes. Le premier, la position centrale x_a du mot a, définit le point de l'espace autour duquel la représentation du terme va fluctuer. Le second terme définit la transformation opérée sur cette représentation en fonction du temps – et plus particulièrement en fonction des groupes auxquels le mot appartient : $\mathcal{C}$ est l'ensemble des groupes, ρ_{ac} est la probabilité avec laquelle le mot a appartient au groupe c, f_c est une fonction qui définit l'influence du groupe c sur la représentation des mots à un temps t (probabilité du groupe d'être actif au temps t), et finalement, g_c est une fonction qui spécifie comment transformer la représentation x_a d'un mot a. Les paramètres ρ sont appris (voir 4.3), et la représentation intiale x_a est donnée par word2vec.

1. En suivant l'article de Mikolov, nous avons choisi une valeur de $k = 5$

Nous proposons plus loin deux modèles de transformation et utilisons par contre dans les deux cas une même fonction f_c, définie formellement par :

$$f_c(t) = \exp^{-|t_c - t|\tau} \quad [5]$$

où τ est un paramètre définissant l'influence temporelle du groupe[2] et t_c est le temps associé au groupe c. Le paramètre τ contrôle d'une certaine façon la granularité du temps considérée pour un groupe donné. Plus τ est grand, plus l'influence temporelle est concentrée autour de t_c.

Dans cet article, nous fixons τ à une valeur qui correspond aux événements principaux présents dans le corpus étudié, mais ce paramètre pourrait être appris. Outre l'apprentissage du paramètre τ, d'autres modèles temporels pourraient être considérés, pour pouvoir par exemple capturer des événements cycliques ou brusques, en utilisant des fonctions périodiques.

Dans la suite, nous définissons deux types de transformations qui définissent l'influence des groupes dans l'espace des représentations.

4.1. *Modèle d'attraction*

Le premier modèle que nous proposons est basé sur l'idée que la représentation d'un mot est "attirée" vers la représentation thématique u_c du groupe c. Formellement, la transformation est définie comme :

$$g_c(x_a) = u_c - x_a \quad [6]$$

où u_c est le vecteur associé au groupe $c \in \mathcal{C}$. En intégrant [6] dans [4], cela nous donne

$$x_a^{(t)} = x_a + \sum_{c \in \mathcal{C}} \rho_{ac} \exp^{-|t_c - t|\tau} (u_c - x_a) \quad [7]$$

Dans le cas où ρ_{ac} est égal à 1, et en ignorant l'influence des autres groupes, la représentation du mot a au temps t_c est donc exactement égale à u_c – et cette représentation retourne de façon continue à x_a quand la différence entre t et t_c augmente.

4.2. *Modèle de translation*

Notre second modèle suppose que les mots appartenant à un groupe sont déplacés de manière similaire dans l'espace thématique. Formellement, la transformation est définie par :

$$g_c(x_a) = u_c \quad [8]$$

2. Nous n'apprenons pas τ dans ce travail préliminaire

En intégrant [8] dans [4], cela nous donne

$$x_a^{(t)} = x_a + \sum_{c \in \mathcal{C}} \rho_{ac} \exp^{-|t_c - t|\tau} u_c \quad [9]$$

Ce modèle est moins adapté à la trajectoire des mots car celui-ci n'augmente que de manière indirecte la similarité (au sens du produit scalaire) entre deux différents mots, contrairement au modèle d'attraction. C'est toutefois un modèle plus simple qui pourrait être appris plus facilement.

4.3. *Apprentissages des groupes*

Afin d'apprendre la représentation des groupes (dans l'espace et le temps), ainsi que le degré d'appartenance des mots aux différents groupes ρ, nous nous basons sur la même fonction objectif que word2vec (section 3) que nous modifions afin de prendre en compte le temps :

$$\mathcal{L} = \prod_{(a,b,t) \in \mathcal{D}} p(b \mid a, t) \quad [10]$$

où l'ensemble d'apprentissage $\mathcal{D}$ correspond à des triplets (a, b, t) pour lesquels le mot b apparait à proximité de a au temps t. Par exemple, en prenant la figure 1, les triplets (a,b,t_1), (a,a,t_1), (k,f,t_2) et (k,e,t_2) appartiennent à $\mathcal{D}$.

Comme dans (Mikolov *et al.*, 2013), la probabilité $p(b \mid a, t)$ dépend de la similarité entre deux mots dans l'espace vectoriel, à savoir

$$p(b \mid a, t) \propto \exp(y_a^{(t)} \cdot x_b^{(t)}) \quad [11]$$

où la représentation de deux mots est définie par le modèle d'attraction ou de translation.

Les deux formules précédentes montrent qu'il est possible d'apprendre tous les paramètres décrits plus haut – les paramètres des groupes (u_c et t_c), le degré d'appartenance ρ_{ac} de chaque mot a au groupe c – en utilisant le principe de la maximisation de la vraisemblance, ou plus précisément le critère basé sur un échantillonnage négatif. Dans ce travail, nous pré-calculons la position centrale x_a et y_a de chaque mot en utilisant le logiciel `word2vec`[3] afin de réduire le nombre de paramètres à apprendre – mais il est tout à fait possible d'optimiser les deux jeux de paramètres en même temps.

5. Application : Clustering

Une fois les représentations des mots et des groupes apprises, notre modèle peut être utilisé pour regrouper les documents, ce qui est utile à des fins de visualisation et

3. `https://code.google.com/p/word2vec/`

d'évaluation. Dans cet article, nous utilisons les groupes pour évaluer notre modèle en terme de regroupement de documents (*clustering*).

En utilisant l'hypothèse du "Naïve Bayes", i.e. en supposant que l'occurrence d'un mot et le temps sont indépendants si le groupe est connu, et en supposant que chaque groupe a la même probabilité *a priori* :

$$p(c|d) = \frac{p(c)p(t_d|c)\prod_{a\in d} p(c|a)p(a)/p(c)}{p(d)} \propto f_c(t_d)\prod_{a\in d}\rho_{ac}$$

où d est un document associé à un temps t_d, et où le produit est défini sur l'ensemble des mots qui apparaissent dans le document d. Nous assignons ensuite le document d au groupe c_d qui a la plus grande probabilité $p(c|d)$.

5.1. *Protocole expérimental*

Nous avons utilisé une collection de 12 millions de micro-blogs (Yang et Leskovec, 2011), un sous-ensemble de micro-blogs de Twitter de juin 2009, que nous avons téléchargé en utilisant l'API de twitter[4]. Ces documents peuvent être écrits avec n'importe quelle langue, et traiter de n'importe quel thème. Nous avons tout d'abord nettoyé le jeu de données de la façon suivante. Nous avons enlevé les micro-blogs avec moins de 8 mots, ce qui a réduit l'ensemble à 10 millions de documents. Pour ceux restant, nous avons enlevé les nombres, la ponctuation, les hyperliens, les liens utilisateurs (*@user*) et étiquettes (les *hashtags* de la forme `#tag`).

Nous avons gardé l'association entre les micro-blogs et les étiquettes afin d'évaluer les groupes trouvés automatiquement par les différents algorithmes, en supposant qu'une étiquette correspond à un groupe donné.

Les groupes des micro-blogs ont été calculés en suivant la méthodologie décrite dans la section 5 pour notre modèle, ou en utilisant l'algorithme des k-moyennes. Des expériences préliminaires avec le modèle (Wang *et al.*, 2012) ont montré qu'il était difficile d'obtenir des résultats satisfaisants avec l'implémentation actuelle de cet algorithme[5]. Pour les k-moyennes, nous avons utilisé une représentation de chaque micro-blog dans l'espace de représentation thématique donné par `word2vec`, en sommant les vecteurs représentant chaque mot contenu dans le micro-blog.

Les groupes de références ont été constitués en faisant l'hypothèse qu'un groupe est défini par les micro-blogs contenant une étiquette donnée. Étant donné le grand nombre d'étiquettes dans ce corpus, nous avons sélectionné des sous-ensembles de taille réduite (environ 100 étiquettes), en suivant la méthodologie suivante. Tout

4. Les identifiants des tweets sont disponibles sur demande

5. Le corpus sur lequel il était évalué est bien plus petit ; nos premières tentatives pour adapter l'algorithme à une grande collection de tweets ont été infructueuses

d'abord, les étiquettes présentes dans moins de 500 micro-blogs ont été enlevées. Ensuite, nous avons généré les ensembles suivants :

Random Un ensemble choisi aléatoirement (probabilité uniforme) ;

Top L'ensemble des étiquettes les plus fréquentes ;

H.C. Un ensemble d'étiquettes choisies à la main parmi les plus fréquentes – en essayant de favoriser celles correspondant à un événement couvrant une courte période (quelques jours) ;

G.I. 0.80, 0.85 et 0.90 Les étiquettes pour lesquelles l'index de Gini (Gini, 1912) est au dessus d'un certain seuil. L'index de Gini a été calculé en se basant sur la fréquence d'occurrence d'une étiquette pour chaque jour. Plus une étiquette apparait de manière uniforme au cours des jours, plus son index de Gini est bas, et plus sa distribution est biaisée, plus son index de Gini est haut. Nous avons choisi de manière empirique les seuils de 0.8, 0.85 et 0.90 qui correspondent à des distributions très "inégalitaires" (quelques jours).

Les trois premiers ensembles (Random, Top, HC) ont plus de chance de favoriser l'algorithme des K-moyennes, car ils correspondent à des thèmes qui ont été traités tout au long de la période correspondant au mois de juin, contrairement aux derniers (GI 0.80, 0.85 et 0.90) qui correspondent plus à des thèmes s'étendant sur quelques jours, et qui *a priori* doivent être mieux détectés par notre modèle.

Pour mesurer la qualité des groupes trouvés, nous avons utilisé la mesure V (Rosenberg et Hirschberg, 2007), une mesure qui compare les groupes trouvés et ceux de référence en calculant la moyenne harmonique de deux mesures à valeurs dans $[0, 1]$:

1) L'homogénéité, qui est liée à l'entropie conditionnelle $H(C|K)$ d'un groupe C étant donné les classes K, est d'autant plus grande que le groupe trouvé est "pur", i.e. qu'il ne contient que des documents avec les mêmes étiquettes ;

2) La complétude, qui est liée à l'entropie conditionnelle $H(K|C)$, est d'autant plus grande que les documents associés à la même étiquette sont dans le même groupe trouvé.

Tout comme la précision et le rappel, augmenter l'homogénéité a tendance à faire diminuer la complétude, et vice-versa.

5.2. *Apprentissage*

Tous les modèles sont initialisés avec un nombre de groupes K fixé à 1000 afin que les modèles puissent détecter un nombre suffisant de thèmes. Les paramètres initiaux des groupes ont été initialisés de manière aléatoire, en utilisant une distribution uniforme pour les temps associés aux groupes (t_c). Nous avons utilisé un espace de dimension 100 pour les modèles de représentations de mots. La représentation centrale

des mots x_a a été pré-calculée en utilisant word2vec[6] sur le même jeu de données, et n'a pas été modifiée par notre algorithme afin de réduire le temps de calcul et d'obtenir des solutions plus stables. Notons que nous n'avons utilisé qu'une seule représentation par mot, contrairement à (Mikolov *et al.*, 2013) où deux représentations sont utilisées pour chaque mot.

Pour apprendre le modèle, nous avons suivi la méthodologie de (Mikolov *et al.*, 2013), en utilisant un échantillon d'exemples "négatifs" (*Negative Sampling Strategy*) :

$$\sum_{(a,b,t)\in\mathcal{D}} \left(\underbrace{\log \sigma \left(x_a^{(t)} \cdot x_b^{(t)} \right)}_{u_p} + \underbrace{k\mathbb{E} \log \sigma \left(-x_a^{(t)} \cdot x_c^{(t)} \right)}_{u_n} \right) \quad [12]$$

où u_p représente les exemples "positifs" (observés) et rapproche la représentation des mots a et b qui co-occurrent ; u_c représente les exemples "négatifs", et éloigne la représentation de a et c. Dans notre cas, les mots vont se rapprocher (ou s'éloigner de manière similaire) en (1) de façon locale, en augmentant leur appartenance aux mêmes groupes (2) de façon globale, en déplaçant les temps associés aux groupes t_c vers les moments où les mots qui appartiennent à ces groupes co-occurrent.

Contrairement à (Mikolov *et al.*, 2013), notre but est de calculer la représentation des groupes et non des mots. Nous utilisons une descente de gradient, où nous mettons à jour de manière alternée les appartenances aux groupes (ρ) et les centres des groupes (u_c et t_c), en utilisant un pas de gradient de 3e-3 (pour u_c et ρ) et 3e3 (pour t_c) qui ont été déterminés expérimentalement. Nous avons utilisé une valeur de tau de 5e-5, ce qui veut dire, étant donné que le temps est mesuré en secondes, que f_c a une valeur de respectivement 0.84 et 0.01 quand la différence entre t et t_c est respectivement une heure et un jour.

5.3. *Résultats*

Dans la table 1, nous donnons la mesure V pour les différents jeux d'étiquettes sélectionnés et modèles. Comme attendu, nos modèles se comportent mieux lorsque les étiquettes choisies correspondent (explicitement ou implicitement) à des événements qui couvrent peu de jours. La différence avec l'algorithme des k-moyennes est très importante pour ces quatre jeux d'étiquettes, ce qui signifie que les groupes trouvés, même s'ils ne sont que le sous-produit du modèle de trajectoire de mots, sont capables de capturer de manière satisfaisante les événements ainsi que les mots qui les décrivent.

Au niveau de nos modèles, le modèle d'attraction obtient des résultats qui sont meilleurs (0.03 à 0.05 points de différence), sauf dans le cas d'étiquettes choisies aléatoirement ou suivant leur fréquence. Cela correspond à l'hypothèse que nous avions

6. `http://word2vec.googlecode.com`

posée – la similarité des mots dans l'espace thématique est mieux prise en compte lorsque les mots convergent tous vers le centre du groupe ; ce qui était moins évident était que cela aurait une influence sur la qualité des groupes trouvés. Notons que cette différence s'inverse lorsque les jeux d'étiquettes ne sont plus ceux favorisant les modèles temporels.

Étiquettes \ Modèle	Attraction	Translation	K-moyennes
G.I. > 0.80	0.48	0.45	0.43
G.I. > 0.85	0.52	0.48	0.44
G.I. > 0.90	0.56	0.51	0.47
H.C.	0.47	0.44	0.43
Top 100	0.16	0.18	0.30
Random 100	0.15	0.18	0.29

Tableau 1. *Mesure-V pour différents modèles et jeux d'étiquettes – le meilleur résultat pour chaque ensemble d'étiquettes est présenté avec un fond vert*

Afin de voir plus en détail le comportement des modèles, les tableaux 2 et 3 donnent respectivement les valeurs de l'homogénéité et de la complétude, et montrent l'algorithme des k-moyennes a une meilleure homogénéité (sur tous les jeux d'étiquettes) mais une complétude bien plus faible (surtout pour les jeux d'étiquettes correspondant à des événements courts). Nous pensons que cela est dû au fait que l'algorithme des k-moyennes n'utilise pas l'information temporelle, et peut potentiellement classer des micro-blogs de mêmes dates dans des groupes différents, alors que nos modèles, qui utilisent l'information temporelle, vont avoir tendance à mettre dans les même groupe des documents écrits à la même date.

Étiquettes \ Modèle	Attraction	Translation	K-moyennes
G.I. > 0.80	0.52	0.45	0.56
G.I. > 0.85	0.58	0.51	0.59
G.I. > 0.90	0.65	0.55	0.65
H.C.	0.49	0.43	0.56
Top 100	0.19	0.21	0.43
Random 100	0.21	0.19	0.47

Tableau 2. *Homogénéité pour différents modèles et jeux d'étiquettes – le meilleur résultat pour chaque ensemble d'étiquettes est présenté avec un fond vert*

Nous avons également regardé, pour le modèle d'attraction, l'influence de la sélection du paramètre τ. Le tableau 4 donne les mesures d'homogénéité pour le jeu d'évaluation G.I.> 0.90 et le modèle d'attraction. Un τ plus petit implique que les groupes influencent la représentation sur moins de jours ; cela entraîne une homogénéité moindre. D'un autre côté la complétude augmente, mais sans compenser la perte en homogénéité. Dans nos expériences, un τ trop petit entraîne une chute significative

Étiquettes \ Modèle	Attraction	Translation	K-moyennes
G.I. > 0.80	0.45	0.45	0.34
G.I. > 0.85	0.47	0.47	0.34
G.I. > 0.90	0.49	0.48	0.36
H.C.	0.43	0.44	0.35
Top 100	0.14	0.16	0.24
Random 100	0.13	0	0.21

Tableau 3. *Complétude pour différents modèles et jeux d'étiquettes – le meilleur résultat pour chaque ensemble d'étiquettes est présenté avec un fond vert*

des performances. Il est donc important, dans les travaux futurs, d'apprendre automatiquement ce paramètre.

mesure\τ	5e-5	1e-5
homogénéité	0.65	0.54
complétude	0.49	0.53

Tableau 4. *Influence de différentes valeurs de τ (le jeu d'évaluation G.I.> 0.90 et le modèle d'attraction)*

Date	Nom	Mots
26 juin 2009	Mort de M. Jackson	jackson, mj, michael, rt, thriller, pop, jackson's, tribute, king, death, brown, farrah, rip, men, died, mj's, dead
29 juin 2009	Hommages à M. Jackson	awards, was, bet, rt, just, chris, man, tribute, mj, joe, brown, jackson, drake, looks, sing, performance, stage, tiny, out, lil, died, show, fan, now, off, king
21 juin 2009	Élections en Iran	ppl, violence, panic, iran, protest, regime, tehran, reader, baby, iranian, seven, police, father's, protesters, symbol, neda, gov, killed, changing, dignity, revolution

Tableau 5. *Groupes trouvés par la méthode proposée en section 5. Les mots avec le ρ le plus haut sont donnés plus haut (après avoir enlevé les mots sans sémantique). Les noms des groupes ont été déterminés manuellement.*

Finalement, nous donnons quelques exemples de groupes trouvés dans le tableau 5. Les groupes ont été sélectionnés en choisissant ceux qui étaient les plus importants (en terme de somme des ρ sur l'ensemble des mots). Ensuite, les mots dont l'appartenance au groupe était la plus forte ont été sélectionnés pour chaque groupe. Les deux premiers groupes sont tous deux liés à un aspect de la mort de Michael Jackson[7] - le

7. Il est mort le 25 juin 2009

premier contenant des micro-blogs datant du jour de sa mort, et les autres décrivant les "BET awards" qui se sont transformés en une commémoration de la mort de l'artiste.

6. Conclusion

Dans cet article, nous avons présenté un ensemble de modèles qui permettent d'apprendre des représentations de mots qui varient dans le temps en fonction de leur appartenance à des groupes (clusters) : chaque groupe définit une transformation de la représentation du mot dans l'espace qui est une déformation continue en fonction du temps. Nous avons ensuite détaillé deux modèles : dans le premier les mots sont attirés vers le centre du groupe, et dans le second la représentation des mots est modifiée par une translation. L'avantage de ce type d'approche est qu'il n'est plus nécessaire d'utiliser un temps discret (ce qui peut poser des problèmes si les événements ont des durées très différentes) et d'estimer la représentation d'un mot à chaque pas de temps (ce qui est coûteux en terme d'espace)

Les expériences que nous avons conduites sur un corpus de micro-blogs (issus de Twitter) ont montré que nos modèles détectaient mieux les thèmes, tels que définis par des étiquettes (*hashtags*), que l'algorithme des k-moyennes, à partir du moment où ces thèmes n'étaient pas constamment actifs dans le corpus.

Dans nos travaux futurs, nous apprendrons directement la fonction d'influence temporelle. Nous nous intéresserons également à l'utilisation de ces représentations temporelles pour d'autres tâches que le clustering, comme par exemple pour voir comment des concepts évoluent dans le temps. Il serait également intéressant d'utiliser ce type de modèles pour d'autres problèmes où une représentation distribuée d'entités est utilisée.

Remerciements

REQUEST, projet investissement d'avenir, 2014-2017.

7. Bibliographie

Beykikhoshk A., Arandjelović O., Venkatesh S., Phung D., « Hierarchical Dirichlet Process for Tracking Complex Topical Structure Evolution and Its Application to Autism Research Literature », *Advances in Knowledge Discovery and Data Mining*, Springer, 2015.

Blei D. M., Ng A. Y., Jordan M. I., « Latent dirichlet allocation », *Journal of Machine of Machine Learning*, vol. 3, p. 993-1022, 2003.

Chen Y., Amiri H., Li Z., Chua T., « Emerging topic detection for organizations from micro-blogs », *ACM SIGIR '13*, 2013.

Deerwester S., Dumais S. T., Furnas G. W., Landauer T. K., Harshman R., « Indexing by latent semantic analysis », *Journal of the American Society for Information Science*, vol. 41, n° 6, p. 391-407, September, 1990.

Gini C., « Variabilità e mutabilità », 1912.

Hall D., Jurafsky D., Manning C. D., « Studying the History of Ideas Using Topic Models », *EMNLP*, 2008.

Liu N., « Topic Detection and Tracking », *Encyclopedia of Database Systems*, 2009.

Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J., « Distributed Representations of Words and Phrases and their Compositionality. », *NIPS'14*, vol. cs.CL, p. 3111-3119, 2013.

Rosa K. D., Shah R., Lin B., Gershman A., Frederking R., « Topical Clustering of Tweets », *SIGIR SWSM Workshop*, 2011.

Rosenberg A., Hirschberg J., « V-measure : A conditional entropy-based external cluster evaluation measure », *EMNLP-CoNLL*, p. 410-420, 2007.

Saha A., Sindhwani V., « Learning evolving and emerging topics in social media : a dynamic NMF approach with temporal regularization », *ACM WSDM'12*, 2012.

Wang X., McCallum A., « Topics over time : a non-Markov continuous-time model of topical trends », *Proceedings of the 12th ACM SIGKDD*, 2006.

Wang Y., Agichtein E., Benzi M., « TM-LDA : efficient online modeling of latent topic transitions in social media », *ACM KDD '12*, 2012.

Yang J., Leskovec J., « Patterns of temporal variation in online media », *WSDM*, ACM, 2011.