Utilisation conjointe LDA et Word2Vec dans un contexte d'investigation numérique

Julien Maitre, Michel Menard, Guillaume Chiron, Alain Bouju

Utilisation conjointe LDA et Word2Vec dans un contexte d'investigation numérique

Julien Maitre[*,**] Michel Menard[*,***]
Guillaume Chiron[*,****] Alain Bouju[*,♯]

[*]L3I, Université de La Rochelle, Avenue Michel Crépeau 17042 La Rochelle
[**]julien.maitre@univ-lr.fr, [***]michel.menard@univ-lr.fr
[****]guillaume.chiron@univ-lr.fr, [♯]alain.bouju@univ-lr.fr

Résumé. L'étude présentée dans cet article s'inscrit dans le contexte du développement d'une plateforme d'analyse automatique de documents associée à un service caché lanceurs d'alerte focalisé sur la révélation de faits/événements/actions en lien avec des problématiques environnementales. Dans le but de traiter de manière automatique les documents textuels révélés par un lanceur d'alerte et portant sur un ou plusieurs faits relatifs à un événement déclencheur, nous proposons de développer un framework d'investigation qui doit répondre au besoin qu'ont les journalistes/politiques/juristes de se munir d'outils d'investigation (extraction, vérification, corrélation) et de représentation de l'information (synthèse, aide à la décision). Il a pour but de faciliter les expertises indépendantes, protéger les lanceurs d'alerte et aider à la détection des signaux faibles. Cet article se focalise plus particulièrement sur le clustering thématique multi-niveaux de documents et l'extraction des indicateurs caractéristiques et significatifs des thèmes. Nous étudions notamment la pertinence d'évaluer une approche s'appuyant sur du comptage de mots par une méthode récente de type "word embedding", *word2vec*. Nous proposons d'évaluer les partitions obtenues grâce à un indice de cohérence sur la collection de mots représentative de chaque thème obtenu. Deux algorithmes sont proposés. Le premier estime le nombre de thèmes le plus pertinent, et extrait ainsi sur ce niveau la collection de mots pour chacun des thèmes trouvés. Le second propose d'extraire les meilleurs collections de mots potentiellement présentes sur des niveaux différents.

1 Introduction

Une problématique majeure actuelle porte sur notre capacité à prendre des décisions éclairées devant l'augmentation drastique des signaux délivrés par toujours plus de moyens d'information. Des phénomènes de saturation des capacités de nos systèmes de traitement conduisent à des difficultés d'interprétation ou même à refuser les signaux précurseurs de faits ou d'événements. L'utilité de la prise de décision contrainte par des nécessités temporelles oblige un traitement rapide de la masse d'information. Etre capable de détecter dans un délai imposé, les bons signaux porteurs de l'information utile dans un contexte de stratégie d'anticipation,

s'avère être un challenge devenu permanent pour de nombreux acteurs économiques. Il est donc nécessaire de développer, sous la forme de plateformes d'investigation (cf. Figure 1), de nouveaux services d'aide à la décision pour les politiques et les organisations en charge de ces activités. Les prises de décision, qui doivent portées aussi bien sur la crédibilité de la source d'information que sur la pertinence des informations révélées relatives à un événement, nécessitent des algorithmes robustes de détection des signaux faibles, d'extraction et d'analyse de l'information portée par ces derniers, d'ouverture sur un contexte informationnel plus large.
Nous proposons de porter notre action sur deux points essentiels : la détection des signaux faibles et l'extraction de l'information véhiculée par ces derniers. Notre objectif concerne donc la détection de signaux précurseurs dont la présence attenante dans un espace de temps et de lieux donnés anticipe l'avènement d'un fait observable. Cette détection est facilitée par l'information précoce délivrée par un lanceur d'alerte sous la forme de documents. Ils exposent des faits avérés, unitaires et ciblés, mais aussi partiaux, relatifs à un événement déclencheur. Le lanceur d'alerte délivre une information non encore décelable/apparente sur les réseaux sociaux et spécialisés. Elle permet de dessiner le contour des signaux faibles à venir sur les réseaux, facilitant ainsi leur détection et l'extraction de l'information portée par ceux-ci.

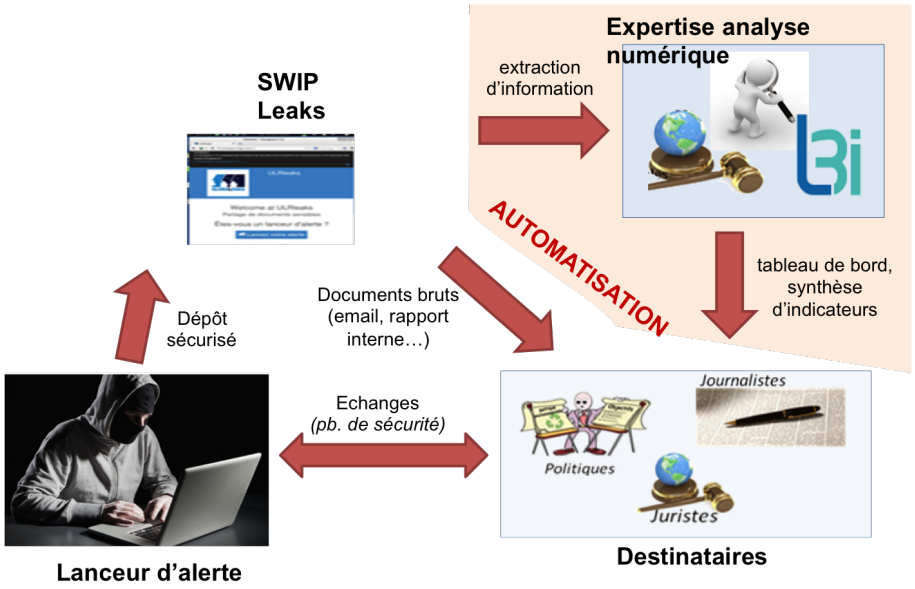

FIG. 1 – *Plateforme d'investigation. Elle doit répondre au besoin réel qu'ont les journalistes/politiques/juristes de se munir d'outils d'investigation (extraction, vérification, corrélation) et de représentation de l'information (synthèse, aide à la décision). Elle a donc pour but de faciliter les expertises indépendantes, protéger les lanceurs d'alerte et aider à la détection des signaux faibles.*

La procédure d'investigation proposée repose donc sur la détection des signaux faibles présents sur les réseaux. Elle combine algorithmes de fouille de données et visualisation analytique. Elle est facilitée par la connaissance des patterns révélés par le lanceur d'alerte. L'in-

formation est estimée à partir des indicateurs révélés par le lanceur d'alerte et des données portées par les signaux faibles (cf. Figure 2). Les smart data, révélées par le lanceur d'alerte, permettent de mieux cibler le data mining lors des phases de détection des signaux faibles et d'exploration sur les réseaux. Pour le développement du framework d'investigation, trois actions sont donc entreprises :

— Action 1 : Analyse automatique de contenus avec un minimum d'*a priori*. Identification des informations pertinentes. Indicateur de cohérence des thèmes obtenus ;
— Action 2 : Agrégation de connaissances. Enrichissement de l'information. Détection des signaux faibles ;
— Action 3 : Visualisation analytique. Mise en perspective de l'information par la création de représentations visuelles et de tableaux de bord dynamique.

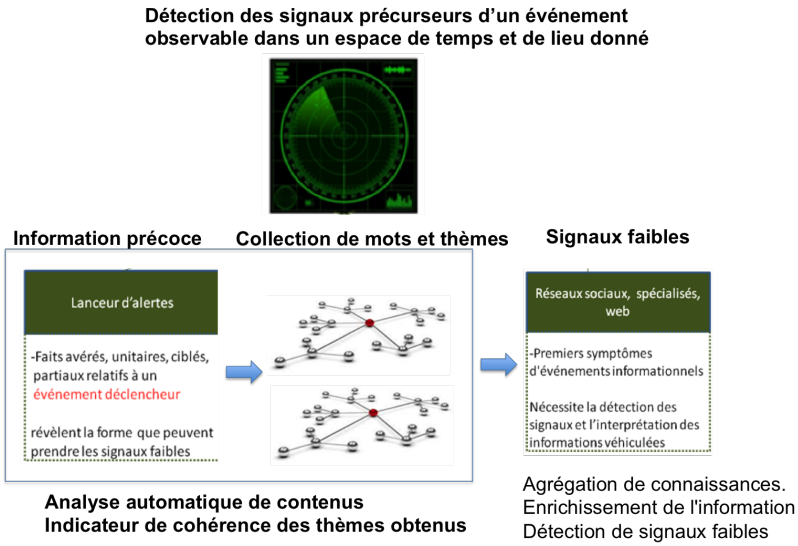

FIG. 2 – *Stratégie de détection des signaux faibles. Elle passe par l'analyse de l'information précoce apportée par un lanceur d'alerte et l'extraction des collections de mots associées aux thèmes découverts. Ces informations permettent ensuite de mieux cibler la phase de data mining pour la détection des signaux faibles.*

Cet article s'inscrit dans la première action. Afin de traiter de manière automatique les documents textuels révélés par le lanceur d'alerte et portant sur un ou plusieurs faits relatifs à l'événement déclencheur, nous développons des outils d'analyse afin de :

— regrouper les documents portant sur un même fait/thème ; le clustering *LDA* (Latent Dirichlet Allocation) permet, à partir de vecteurs descripteurs construits sur les documents, de relier ensemble avec un minimum d'*a priori* tous les documents relatifs à un même thème. Ces thèmes, que nous supposerons relatifs à l'événement déclencheur, sont découverts simultanément grâce au clustering multi-niveaux

— d'évaluer la qualité des partitions obtenues grâce à un indice de cohérence sur la collection de mots représentative de chaque thème obtenu. Deux algorithmes sont proposés. Le premier estime le nombre de thèmes le plus pertinent, et extrait ainsi sur ce niveau la collection de mots pour chacun des thèmes trouvés. Le second propose d'extraire la meilleure collection de mots pour chaque thème, celle-ci pouvant être potentiellement présente sur des niveaux différents. Ces mots et leurs attributs sont les indicateurs recherchés (lexique de descripteurs textuels). Ils seront utilisés par la suite lors des requêtes pour enrichir ce premier niveau d'information.

2 Clustering et text mining

La nécessité grandissante du traitement rapide de l'information conduit au développement de nombreux algorithmes utilisant diverses approches pour traiter les données. Des méthodes de traitement et d'extraction efficaces, comme *LDA* ou dérivées du "word embedding" sont régulièrement utilisées.
Le problème qui nous intéresse dans cette étude est celui de l'évaluation de l'efficacité du modèle *LDA*. Ce dernier catégorise les documents en un nombre de thèmes défini *a priori*. Afin d'améliorer la séparation en thèmes des documents, il est nécessaire (1) de faire varier ce paramètre, (2) d'estimer le niveau de partitionnement le plus pertinent, et (3), d'estimer dans l'arbre de profondeur la meilleures collection de mots représentative d'un thème. Pour cela nous nous appuyons sur une méthode récente de type "word embedding", *Word2Vec*. Celle-ci s'appuie sur une méthode d'apprentissage automatique issue du deep learning. Elle permet de représenter un mot par un vecteur dans un but d'analyse sémantique. Ainsi deux mots dans des contextes similaires ont des vecteurs proches. Cette approche s'avère donc complémentaire des méthodes s'appuyant sur le comptage de mots dans un document. Elle projette les mots dans un espace de vecteur en fonction du contexte local d'une phrase, au contraire du modèle *LDA* qui trie les mots en fonction de leurs probabilités dans les thèmes.
Nous commençons d'abord par décrire le fonctionnement de *LDA* ainsi que sa mise en oeuvre. Nous présentons ensuite l'approche *Word2Vec* et ce qu'elle apporte dans l'optimisation de *LDA*. Nous terminons par une présentation des deux algorithmes proposés et une discussion sur les résultats.

2.1 Quelques méthodes algébriques de représentation d'un document

Parmi l'ensemble des techniques de traitement des langues naturelles, l'analyse sémantique latente de Deerwester et al. (1990) (ou LSA) fait figure de pionnière. Cette technique décrit les relations entre les documents et les mots qu'ils contiennent. Une version probabiliste pLSA a servi d'inspiration pour le modèle LDA. pLSA de Hofmann (1999) intègre des techniques statistiques pour le traitement des mots où leurs composantes peuvent être considérées comme des représentations de «sujets». Chaque mot est ainsi généré à partir d'un seul sujet. Des variantes à LDA existent comme la méthode hiérarchique hLDA proposé par Blei et al. (2004).

2.2 Latent Dirichlet Allocation

Le modèle *LDA* est une méthode probabiliste générative de mots proposée par Blei et al. (2003) dont le but est découvrir les thèmes sous-jacents à un ensemble de documents. Chacun d'eux est modélisé par un mélange de thèmes générateur des mots du document. *LDA* est un modèle Bayésien à trois couches (cf. Figure 3). Elle utilise l'approche "Bag of Word" qui traite chaque document d du corpus D défini par $(\mathbf{w_1}, ..., \mathbf{w_D})$ comme un N-uplet de mots, $\mathbf{w_d} = (w_1, ..., w_N)$. A chaque mot $w_{(d,n)}$ est alors associé un thème représenté par la variable $z_{(d,n)}$. θ_d représente la distribution de thèmes du document d. Des hyperparamètres, α et η, définissent l'*a priori* sur θ et β où β_k décrit la distribution du thème k.

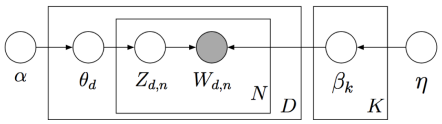

FIG. 3 – *Modèle graphique de* LDA *(Blei et al. (2003))*

LDA est un outil classiquement utilisé dans une grande variété de domaines : structuration automatique de corpus de documents, recommandation, génération de rapport de synthèse... Plusieurs recherches ont évalué l'efficacité de *LDA* dans des domaines où de grands volumes de données doivent être structurés thématiquement, notamment pour les réseaux sociaux, Hong et Davison (2010), le web profond, Noor et al. (2013) ou les encyclopédies numériques telle Wikipedia, Hoffman et al. (2010). Un grand nombre d'extensions de *LDA* ont également été proposé, par exemple, pour l'analyse sémantique latente probabiliste, PLSA, en traitement automatique des langues.
Dans la majorité des travaux, les utilisateurs utilisent *LDA* avec un nombre de thèmes *a priori*. Par exemple, Noor et al. (2013) utilise le dataset TEL-8, un ensemble de sources classé en 8 domaines ou bien Hoffman et al. (2010) qui cherche à structurer un ensemble d'articles de Wikipedia en 100 topics.

2.3 Mise en oeuvre. Application de *LDA* sur la base de connaissance Wikipedia

L'encyclopédie numérique Wikipedia français contient 3.9 millions de pages. Pour le choix du corpus et afin de construire une preuve de concept, nous avons choisi d'utiliser les documents de l"encyclopédie numérique Wikipedia français qui contient un nombre conséquent de documents (3.9 millions de pages). L'ensemble D de notre corpus de documents extrait de cette encyclopedie est constitué de 4 catégories génériques : "Economie", "Histoire", "Informatique", "Médecine" (cf. tableau 1). La structure des catégories dans Wikipedia est une arborescence particulière. Une feuille peut en effet se trouver sur plusieurs branches. De ce fait dans la version anglaise de Wikipedia, Bairi et al. (2015) explique qu'une catégorie principale couvre, si l'on explore ses sous-catégories jusqu'à une profondeur de 10, l'ensemble des documents présent dans l'encyclopédie.

	Nombre de pages	Echantillon de pages (0.6%)
Nombre de pages du Wikipedia français	3 987 661	/
Catégorie "Economie"	1 888 801	9 445
Catégorie "Histoire"	1 743 374	8 717
Catégorie "Informatique"	1 030 280	5 152
Catégorie "Médecine"	570 257	2 852

TAB. 1 – *Nombre de pages dans Wikipedia et dans les différentes catégories. Un échantillon de documents dans chaque catégorie est extrait et constitue notre corpus.*

Il existe de forts recouvrements entre les catégories, c'est pourquoi les pages spécifiques à un thème sont certainement peu nombreuses. Pour préserver une durée de calcul raisonnable, nous avons choisi de mener notre étude sur un sous échantillon du corpus, lequel se compose de 0.6% des documents de chaque catégorie. Cela représente un total de 26 000 documents. Chacune d'elle fait plus de 10 Ko. Le Tableau 2 montre le résultat de partitionnement par la méthode LDA lorsque le nombre de thèmes (ou clusters) k est fixé à 4.

	Thème 1	Thème 2	Thème 3	Thème 4
Mots	commune	film	saison	guerre
	ville	album	club	pays
	roi	premier	première	france
	nom	années	premier	général
	église	ans	tour	français
	...	...	...	...

TAB. 2 – *Liste des 5 premiers mots de chaque thème*

Pour rappel, LDA est une méthode de clustering (non supervisée) et ne permet donc pas d'associer une étiquette aux thèmes trouvés. De plus, il n'est pas possible de discerner la cohérence de chaque thème. Pour cela il est nécessaire de définir un indicateur, c'est l'objet de la prochaine section.

3 Word embedding

Le "word embedding" ou en français "plongement de mots" apporte une solution au problème de la dimensionnalité lié à la taille des dictionnaires. Cette approche permet d'une part de représenter les mots d'un dictionnaire par des vecteurs, et d'autre part, de prendre en compte la notion de contexte, facilitant l'analyse sémantique et syntaxique. Son implémentation s'appuie notamment sur les réseaux de neurones comme ceux présentés par Mikolov et al. (2013) qui permettent des estimations de probabilités significativement meilleures que les modèles n-grammes, (Mikolov et al. (2011), Bengio et al. (2003)). Actuellement, grâce à l'accroissement de la puissance de calcul (programmation GPU) et aux approches d'apprentissage profond, des

problématiques difficiles comme la Traduction, l'Analyse de sentiments ou la Reconnaissance vocale ont connu des avancés significatives.

3.1 Mise en oeuvre des indicateurs

Afin de définir notre indicateur, nous utilisons une méthode récente de type "word embedding" introduite par Mikolov et al. (2013), *Word2Vec*.
Dans Moody (2016), l'auteur décrit un algorithme hybride reposant sur les deux approches *LDA* et *Word2Vec*. L'algorithme porte le nom de *lda2vec*. Le vecteur du mots est estimé par *Word2Vec* en utilisant des informations locales représentées par les mots voisins, et des informations globales au corpus de documents apportées par *LDA*.

Notre approche, contrairement à celle de *lda2vec*, a comme objectif la recherche des clusters de mots ayant la plus forte similarité dans le contexte du corpus de documents. Elle s'appuie sur le fait que les mots sont représentés sous la forme de vecteurs caractéristiques des relations contextuelles qui les relient entre eux par l'intermédiaire de leur contexte (de voisinage). Il est alors possible de définir la valeur de similarité entre deux mots. Une valeur proche de 1 indique que les mots sont très proches l'un de l'autre (i.e. contexte semblable) et possède donc un lien sémantique fort. A l'inverse, 0 indique des mots peu employés dans des contextes semblables. Nous utilisons cette valeur de similarité pour construire un indicateur de cohérence. Celui-ci se définit comme la somme des valeurs de similarité de toutes les combinaisons de mots deux à deux dans chacun des thèmes. Il est donné par l'équation (1) et permet ainsi d'analyser la cohérence d'un thème. E est l'ensemble des 100 mots supports du thème, P l'ensemble des k-combinaisons de E et $w2vSim$ la mesure de similarité définie dans *Word2Vec* par Mikolov et al. (2013). Plus la valeur est grande et plus le thème contient des mots régulièrement employés ensembles.

$$ind = \sum w2vSim(P_{k=2}(E)) \tag{1}$$

L'étape suivante consiste à appliquer l'algorithme *LDA* en faisant évoluer le nombre de clusters/thèmes. Nous obtenons ainsi plusieurs partitions calculées sur un nombre de thèmes différents, et qu'il est possible de représenter sous la forme d'une arborescence. Il est à noter que *LDA* ordonne les thèmes découverts lors des différentes itérations dans un ordre aléatoire, une étape supplémentaire est donc nécessaire pour construire cette arborescence (voir ci-après). Les résultats sont présentés sur le tableau 3. k représente le nombre de clusters donné en entrée de *LDA*.

	Thème 1	Thème 2	Thème 3	Thème 4	Thème 5	Thème 6
2 thèmes (k=2)	1367	2469				
3 thèmes (k=3)	1337	1867	2487			
4 thèmes (k=4)	1480	2052	2356	1948		
5 thèmes (k=5)	2104	1633	3284	1921	1416	
6 thèmes (k=6)	2070	3181	2013	1382	2051	1820

TAB. 3 – *Evolution de l'indicateur en fonction du nombre de thèmes k donné en entrée de* LDA

Afin d'évaluer les partitions obtenues, nous proposons deux algorithmes s'appuyant sur l'indicateur précédent. Le premier (décrit en Section 3.2) estime le nombre de thèmes k (i.e. le niveau de l'arborescence) le plus pertinent, et extrait ainsi sur ce niveau, la collection de mots pour chacun des thèmes trouvés. Le second (décrit en Section 3.3) propose d'extraire les meilleurs collections de mots potentiellement présentes sur des niveaux différents.
Afin de construire l'arborescence, il est nécessaire d'évaluer le lien de ressemblance entre des thèmes de niveaux différents. Celui-ci se calcule au moyen d'un indicateur de ressemblance (2). Pour l'ensemble, C, des mots communs w présents à la fois dans un thème de niveau n, T_n, et un thème de niveau $n+1$, T_{n+1}, nous calculons la somme normalisée du produit des probabilités $p_{.,l}$ associés à ces mots commun dans les deux thèmes (cf. Figure 4).

$$R(T_{n+1}, T_n) = \frac{\sum_{l \in C} p_{n,l} \cdot p_{n+1,l}}{\sum_{l \in T_{n+1}} p_{n+1,l}^2} \tag{2}$$

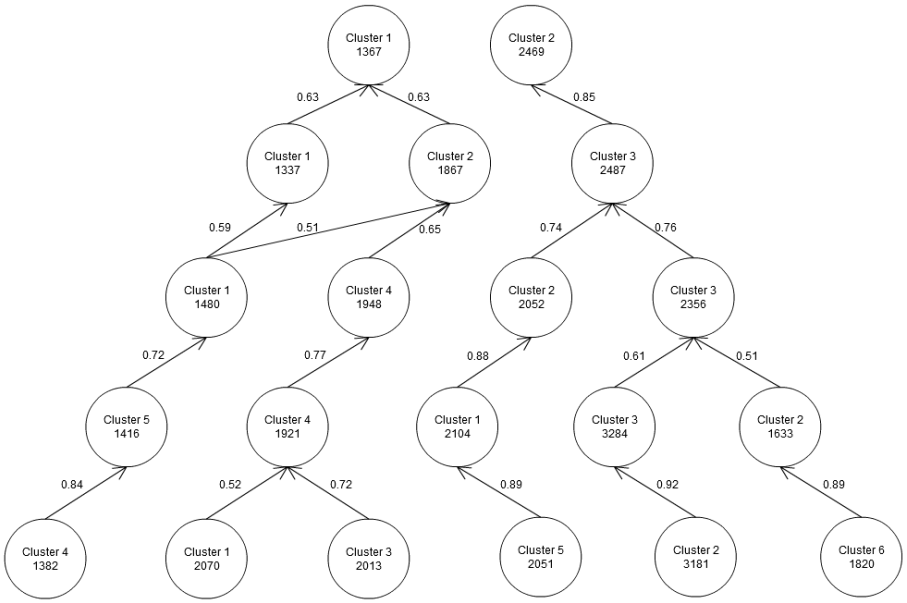

FIG. 4 – *Partitions et arborescence obtenus sur K niveaux (K fixé à 6 dans cet exemple) avec* LDA. *Représentation des liens entre les thèmes au moyen de l'indicateur de ressemblance. Ne sont gardés que les liens dont la valeur est supérieure à 10.*

3.2 Recherche du k le plus pertinent

L'algorithme 1 consiste en la recherche du niveau de l'arborescence donnant les thèmes les plus cohérents au sens de l'indicateur Eq. (1). Sur chaque niveau, nous calculons la valeur minimale de cet indicateur sur l'ensemble des clusters présents sur le niveau. Le niveau retenu (et donc le nombre de clusters pertinent au sens du critère) correspond à celui dont la valeur

mininale est la plus grande. La figure 5 montre les différents clusters. Le cluster dont la cohérence est la plus grande parmi les clusters les moins cohérents de chaque niveau est le thème 1 du niveau 4 avec la valeur 1480 (cf. Figure 5).

3.3 Recherche des clusters les plus pertinents sur l'ensemble de l'arborescence *LDA*

Il est possible d'extraire sur toute l'arborescence les clusters/thèmes les plus pertinents au sens du critère de cohérence, $ind(T)$, et des relations de ressemblance, $R(T_n, T_{n+1})$, entre un cluster de niveau n et de niveau $n+1$. Nous proposons pour cela une méthode de parcours des thèmes dans l'arborescence de manière ordonnée selon le critère ind(T), où chaque thème nouvellement rencontré est retenu comme pertinent et entraine le retrait dans l'arborescence de tous ses thèmes parents ou fils. Les liens de parenté entre les thèmes (décris par R) ne sont considérés qu'au delà d'un seuil fixé arbitrairement à 0,5. L'algorithme 2 formalise ce parcours.

3.4 Interprétation des résultats

Nous avons partitionné les documents en un nombre de thèmes. La recherche des clusters/thèmes les plus pertinents a permis de mettre en évidence la nécessité de faire varier la valeur de k passée en entrée de la méthode LDA. En effet, selon la valeur de k, nous obtenons des clusters de mots avec de fortes valeurs de cohérence et d'autres moins. Le cluster 3 dans la partition à 5 thèmes contient les mots [saison, club, france, match, championnat]. On remarque que ce sont essentiellement des mots en relation avec le sport. Au contraire du cluster 6 dans la partition à 6 thèmes qui contient des mots ayant peu de liens entre eux comme [autres, cas, exemple, système, certains, plusieurs]. On remarque que l'on ne peut pas définir d'étiquette à ce regroupement de mots. En fixant une valeur de seuil (par exemple 2000), nous pouvons identifiter 2 clusters de rejet contenant des mots ayant peu de similarité. Ces clusters de rejet contiennent des groupes de mots qui ont de fortes valeurs de cohérences [commune, ville, région, département, population], mais sont finalement peu nombreux dans le cluster. Un repartitionnement de ces clusters de rejet permettrait de mettre en evidence ces relations faibles.

4 Conclusion

Dans cet article, nous avons proposé une approche pour la recherche de thèmes/faits communs au sein d'un corpus de documents. La combinaison LDA / word2vec telle que nous avons proposé de la mettre en oeuvre permet de s'affranchir du paramètre k (nombre de clusters) pour le partitionnement. Deux directions ont été explorées : 1) un premier algorithme (c.f. Section 3.2) visant à rechercher le nombre de thèmes (paramètre k) entraînant un partitionnement par LDA le plus cohérent possible ; 2) un algorithme (c.f. Section 3.3) qui, de manière plus avancée, combine les meilleurs thèmes renvoyés par LDA sur l'ensemble des partitionnements (ou valeurs de k) testées.
La valeur de seuil fixée pour le parcours et l'élagage de l'arborescence a été arbitrairement fixée à 10. Les valeurs supérieures matérialisent une relation forte entre les thèmes, alors que

les valeurs inférieures peuvent être assimilées à des relations moins évidentes, mais pourtant bien existantes. Actuellement considérés comme des clusters de rejet, ces thèmes et relations aussi infimes soient-elles peuvent éventuellement matérialiser des signaux faibles. Dans le contexte de notre étude sur la détection des signaux faibles et de lancements d'alertes, nous pensons que ces signaux / relations faibles méritent d'être étudiés.
L'information portée par ces derniers devra être corrélé à un contexte informationnel plus large au moyen de phase d'exploration sur les réseaux. Ceci dans un objectif de détection de signaux précurseurs dont la présence attenante dans un espace de temps et de lieux donnés anticipe l'avènement d'un fait observable.

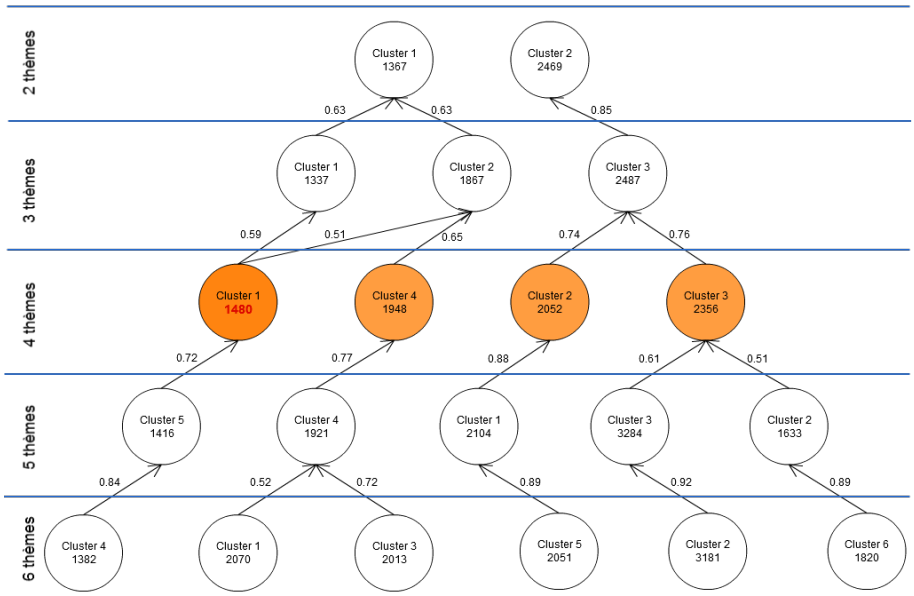

FIG. 5 – *Application de l'algorithme* (1) *sur l'arborescence* LDA *obtenue*

Data : P = Liste des nombres de clusters demandés : {2...K}
Result : bestk = identifiant du k niveau
bestk $\leftarrow$ 0;
bestScorek $\leftarrow$ `Min(LDA(`bestk`))`;
for $k \in P$ **do**
  if `Min(LDA(`k`))` $>$ bestScorek **then**
    bestk $\leftarrow$ k;
    bestScorek $\leftarrow$ `Min(LDA(`k`))`;
  end
end
return bestk

Algorithme 1 : Récupération de l'identifiant du niveau k optimal

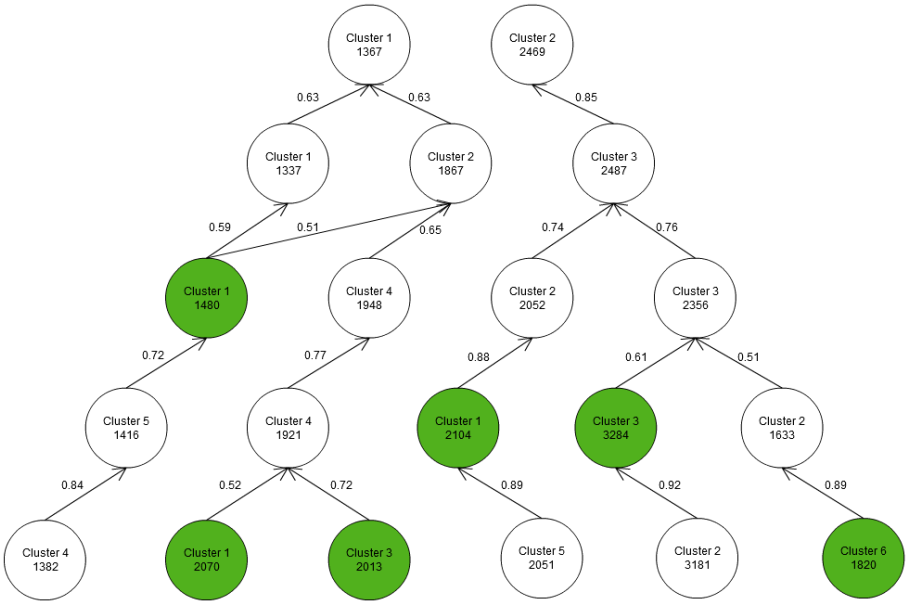

FIG. 6 – *Application de l'algorithme* (2) *sur l'arborescence* LDA *obtenue*

Data : T = Liste des thèmes de l'arborescence *LDA* triés par valeur de cohérence
Result : themesRetenus = Liste des identifiants des thèmes pertinents
themesRetenus $\leftarrow$ {};
while `Taille(`T`)` > 0 **do**
  meilleurCluster $\leftarrow$ `Max(`T`)`;
  themesRetenus $\longleftarrow$ themesRetenus + {meilleurCluster };
  for $t \in$ `Parents(`meilleurCluster`)` **do**
    $T \longleftarrow T - t$;
  end
  for $t \in$ `Fils(`meilleurCluster`)` **do**
    $T \longleftarrow T - t$;
  end
end
return themesRetenus

Algorithme 2 : Récupération des thèmes pertinents dans l'arborescence *LDA*

Références

Bairi, R. B., M. Carman, et G. Ramakrishnan (2015). On the Evolution of Wikipedia : Dynamics of Categories and Articles. *2015 ICWSM Workshop*, 1–8.

Bengio, Y., R. Ducharme, P. Vincent, et C. Janvin (2003). A Neural Probabilistic Language Model. *The Journal of Machine Learning Research 3*, 1137–1155.

Blei, D., T. Griffiths, M. Jordan, et J. Tenenbaum (2004). Hierarchical topic models and the nested Chinese restaurant process. *Advances in neural information processing systems 16*.

Blei, D. M., A. Y. Ng, et M. I. Jordan (2003). Latent Dirichlet Allocation. *Journal of machine learning research : JMLR 3*, 993–1022.

Deerwester, S., S. T. Dumais, et R. Harshman (1990). Indexing by latent semantic analysis. *Journal of the American society for information science 41*(6), 391–407.

Hoffman, M. D., D. M. Blei, et F. Bach (2010). Online Learning for Latent Dirichlet Allocation. *Advances in Neural Information Processing Systems 23*, 1–9.

Hofmann, T. (1999). Probabilistic latent semantic indexing. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '99*, 50–57.

Hong, L. et B. D. Davison (2010). Empirical study of topic modeling in twitter. *Proceedings of the First Workshop on Social ...*, 80–88.

Mikolov, T., G. Corrado, K. Chen, et J. Dean (2013). Efficient Estimation of Word Representations in Vector Space. *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, 1–12.

Mikolov, T., A. Deoras, S. Kombrink, L. Burget, et J. H. Černocký (2011). Empirical evaluation and combination of advanced language modeling techniques. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 605–608.

Moody, C. (2016). Mixing Dirichlet Topic Models and Word Embeddings to Make lda2vec.

Noor, U., A. Daud, et A. Manzoor (2013). Latent Dirichlet Allocation based Semantic Clustering of Heterogeneous Deep Web Sources.

Summary

This paper is related to a wide project aiming at discoving from different streams of information (i.e. daily publication from the Internet), weak signals possibly sent by whistleblowers. The current study presented in this paper tackles the particular problem of clustering topics at multi-levels from multiple documents, and then extracting meaningful descriptors, such as weighted lists of words. In this context, we present a novel idea combining LDA (in charge clustering) and Word2vec (providing a consistency metric regarding the partitioned topics) as potential method for limiting the $apriori$ number of cluster k usually needed in classical partitioning approaches. We proposed 2 implementations of this idea, respectively able to: (1) finding the optimal k for LDA ; (2) gathering the optimal clusters from different levels of clustering.