Exploitation de syntagmes dans la découverte de thèmes

Amaury Delamaire, Michel Beigbeder, Mihaela Juganaru-Mathieu

Univ Lyon, IMT Mines Saint-Etienne, Institut Henri Fayol, Univ Jean Monnet
amaury.delamaire@emse.fr,mbeig@emse.fr,mathieu@emse.fr

RÉSUMÉ. *Le but de cet article est d'étudier l'apport des syntagmes nominaux, verbaux et adjectivaux pour la découverte de thèmes (*topic modeling*). Nous testons l'hypothèse qu'ajouter des syntagmes à la représentation des documents — pour lesquels ne sont traditionnellement considérés que les mots simples — permettrait d'améliorer la qualité d'un modèle de thèmes, en l'occurrence* LDA*. Des différences significatives sont attendues notamment lorsque plusieurs thèmes partagent le même vocabulaire. Nous présentons des résultats sur un corpus catégorisé de 20 000 résumés d'articles scientifiques. Il s'agit d'une étude de cas qu'il conviendrait de reproduire sur un corpus plus conséquent.*

ABSTRACT. *The goal of this paper is to study whether using word syntagms (nominal, adjectival and verbal) is useful for topic modeling. We experiment the hypothesis that adding word syntagms to document representations – for which only single words are usualy considered – would improve a topic model quality,* LDA *in this experiment. Significative differences are expected on topics with common vocabulary. We present results on a categorized corpus of 20 000 scientific article abstracts. This is a case study which should be reproduced for further generalisation.*

MOTS-CLÉS : *Classification non supervisée, TALN, Modèle de thèmes.*

KEYWORDS: *Clustering, NLP, Topic modeling.*

DOI:10.3166/RIA.1.1-14

1. Introduction

Les applications d'apprentissage automatique sont nombreuses tant en milieu industriel qu'au niveau des services web ou des services à la personne, ou encore dans les centres de documentation. Pour le texte, dans la majorité des applications, une technique statistique est mise en place : le nombre d'apparitions d'un «terme» dans un document, une partie de document ou une collection est calculé. Dans ce contexte, «terme» désigne le mot tel qu'il apparaît dans le texte, à savoir une succession de signes isolée par des espaces. Si nous tenons compte de la structure des phrases, le modèle de représentation peut être enrichi de constructions plus élaborées comme les

syntagmes nominaux, verbaux ou adjectivaux. Cette approche pourrait permettre de mieux appréhender le texte et d'obtenir de meilleurs résultats, c'est l'hypothèse que nous testons dans cet article.

Le but des modèles de thèmes est de représenter les thèmes abordés dans un ensemble de documents. Ces modèles permettent, par exemple, de détecter le ou les thèmes abordés dans un nouvel article et de l'indexer en fonction du résultat. Généralement basés sur les mots (unigrammes), nous voulons étudier l'apport de syntagmes extraits par une analyse linguistique, plus précisément tester l'hypothèse qu'ajouter des syntagmes à la représentation des documents améliorera la qualité des résultats. L'évaluation de ces modèles est très largement reliée à celle de la classification automatique. Afin de réaliser notre expérience, nous avons choisi *Latent Dirichlet Allocation* (LDA) (Blei *et al*., 2003) comme modèle de thèmes, parce qu'il est largement testé et non supervisé. L'idée est de construire le modèle sur un corpus donné afin d'évaluer sa sortie d'abord avec des unigrammes, puis d'ajouter des syntagmes à la représentation des documents et d'effectuer les mêmes mesures.

Voici le vocabulaire que nous utiliserons dans cet article :

Catégorie : Sous ensemble de documents du corpus associés à un thème préétabli (par exemple *Mathematics*),

Groupe / Classe : Synonymes pour désigner un paquet de documents identifiés par un algorithme de classification non supervisé, comme équivalent de *cluster* anglais,

Unigramme : Mot simple, sans espace : *réseau*, *de* et *neurones* sont les trois unigrammes de *réseau de neurones*,

ACI : Analyse en Constituants Immédiats (Bloomfield, 1933), elle construit les arbres syntaxiques des phrases d'un texte (cf. section 4.1),

Syntagme : Séquences d'unigrammes correspondant à une fonction donnée dans l'arbre syntaxique de l'ACI.

Nous précisons le cadre de travail et introduisons certains concepts dans la section 2 avant de continuer avec l'état de l'art en section 3. Nous introduisons ensuite les différents protocoles de nos expériences en section 4 avant de poursuivre avec des présentations du corpus d'expérimentation et de nos résultats en section 5.

2. Cadre de travail

Nous nous proposons d'effectuer une étude comparative sur le modèle LDA en exploitant les résultats de l'ACI. De multiples exécutions avec différents prétraitements du texte nous permettront de comparer les modèles de représentation des documents.

Une grande variété de méthodes proposent des solutions de classification, reposant sur des hypothèses diverses. Le modèle LDA tente de décrire des thèmes — dont le nombre est prédéfini — à partir de la distribution du vocabulaire au sein des documents.

Le cœur du modèle LDA travaille sur des représentations de documents sous forme de vecteurs de nombres ; le vecteur de nombres représentant un document donne la fréquence relative de chaque terme du vocabulaire dans ce document. La sortie de LDA est :

1. un ensemble de k thèmes, chaque thème étant une distribution de probabilités sur l'ensemble des éléments constitutifs, qui peut donc être représentée sous forme d'un vecteur de $|\mathcal{V}|$ nombres, $\mathcal{V}$ constituant l'ensemble du vocabulaire,

2. la représentation des documents comme distributions de probabilités sur l'ensemble des k thèmes, un document est donc représenté sous la forme d'un vecteur de k nombres.

LDA pose l'hypothèse que les documents à analyser sont générés par des modèles statistiques correspondants à des thèmes distincts. Ces thèmes correspondent à des variables dites *latentes* que le modèle tente de représenter. Les modèles relatifs aux différents thèmes consistent en une distribution sur l'ensemble des mots du vocabulaire. Pour cela, LDA construit des modèles de co-occurrences évolués et estime la similarité entre ces modèles. Nous proposons ici de modifier le vocabulaire utilisé par le modèle au travers d'une extraction de syntagmes à partir de l'ACI. Différentes adaptations de LDA ont été proposées depuis sa publication, nous en présentons ici plusieurs.

3. Etat de l'art

De nombreux articles illustrent la variété des cas d'application et d'utilisation de LDA : filtre de courriers indésirables (Bíró *et al*., 2008 ; 2009), application sur de courts segments de textes (Phan *et al*., 2008), application sur des mots vides (Arun *et al*., 2009), catégorisation automatique de logiciels à partir du code source (Tian *et al*., 2009).

D'autres auteurs ajoutent des extensions au modèle initial, mais qui ne sont pas linguistiques contrairement à ce que nous nous proposons de faire : (Bolelli *et al*., 2009) ajoutent une couche de diachronie, c'est-à-dire une prise en compte de la date des documents lors de l'analyse afin de mettre en évidence l'émergence des thèmes, (Andrzejewski *et al*., 2011) ajoutent des règles de logique de premier ordre au modèle, (Qian *et al*., 2015) introduisent la notion de multimodalité en vue de la classification d'événements.

(Griffiths *et al*., 2004) modifient le modèle de représentation en filtrant les mots d'un document selon leur importance. L'importance d'un mot prend en compte son étiquette morphosyntaxique et son contexte d'apparition.

(Bíró, Szabó, 2009 ; Liu *et al*., 2011) entrainent un SVM (*Support Vector Machine* (Vapnik, 2013)) à partir des distributions de thèmes de LDA. SVM est un algorithme de classification supervisée, d'où la nécessité d'un corpus d'entrainement étiqueté — ici le résultat de LDA, alors considéré comme vérité terrain. Les auteurs présentent une adaption de la méthode de construction du modèle : au lieu de construire

le modèle sur le corpus complet, ils le construisent de manière supervisée et sur plusieurs corpora (un corpus par groupe). L'expérience tend à prouver que la combinaison de différents modèles construits isolément fournit de meilleurs résultats qu'une seule exécution sur l'ensemble des catégories.

Contrairement à l'expérience que nous présentons, tous ces travaux n'exploitent des documents que sous forme de mots simples.

Des chercheurs ont cependant obtenu une amélioration de la qualité des modèles de thèmes avec un ajout de termes plus complexes, mais se basant soit sur des modèles statistiques de n-grammes et d'alignement de n-grammes, soit sur des thesauri. (Wang *et al*., 2007) présentent une étude comparative entre des variantes de LDA exploitant des n-grammes. Leur solution permet de repérer le bigramme *maison blanche* comme significatif pour le thème *Politique* mais pas pour le thème *Immobilier*. Ils observent un gain en précision significatif sur des tâches de recherche d'informations. (Nokel, Loukachevitch, 2015) et (Nokel, Loukachevitch, 2016) incluent respectivement les bigrammes dans le modèle PLSA (Hofmann, 1999) et des expressions polylexicales issus de thesauri dans le modèle LDA. De même que (Wang *et al*., 2007), ils observent un gain de qualité significatif. (Wallach, 2006) propose une étude comparative entre LDA et d'autres modèles de thèmes et aboutit également à la conclusion que les n-grammes ont un intérêt notable dans les modèles de thèmes.

(Blei, Lafferty, 2009) exploitent quant à eux les n-grammes pour visualiser les thèmes construits. Pour cela, les mots des documents sont assignés à un thème unique après une inférence du modèle. Ces assignations permettent ensuite de trier les n-grammes relativement à leur probabilité d'appartenance à un thème. Contrairement à (Wang *et al*., 2007), les n-grammes de (Blei, Lafferty, 2009) ne sont pas exploités lors de la construction du modèle mais seulement à des fins de visualisation.

Dans la continuité de ces expériences, nous tentons ici d'appliquer des méthodes de traitement automatisé du langage plus complexes telle que l'ACI. Par là même nous tentons de prouver qu'une analyse syntaxique peut compléter les méthodes strictement statistiques.

4. Notre méthode

*4.1. Construction des syntagmes par l'*ACI

Nous postulons que l'hypothèse du sac de mots provoque une perte d'informations que nous tenterons d'atténuer par l'exploitation des syntagmes, notamment lors d'exécutions sur des hiérarchies de catégories.

L'ACI peut se définir comme une analyse linguistique délimitant et organisant des séquences de mots dans un texte à partir de leur nature (cf. section suivante). Elle nous permettra par exemple de considérer *réseau de neurones* comme un seul terme, ce qui devrait aider LDA à distinguer des groupes avec une part de vocabulaire commun, notamment lorsqu'il s'agit de documents techniques. Nous nous proposons d'exploiter

les syntagmes et certains mots filtrés selon leur étiquette morphosyntaxique afin de tenter d'améliorer la qualité des résultats. Deux jeux seront à distinguer, avec ou sans filtre sur les étiquettes (voir section 5.4). Nous avons retenu les éléments suivants pour notre expérience avec filtrage (cf. section 4.2.4) :

Syntagmes : Nominaux, adjectivaux et verbaux,

Etiquettes : Noms communs, verbes et adjectifs.

Pour notre expérience sans filtrage, les mêmes syntagmes sont extraits mais aucun filtre n'est appliqué sur les étiquettes morphosyntaxiques. Dans le cas d'un syntagme contenu dans un autre syntagme, les deux sont retenus. Nous avons choisi tous les niveaux de syntagmes afin de gagner en similarité au niveau des mots composés, mais également de ne pas en perdre au niveau des unigrammes.
Exemple issu du corpus : la phrase « *The scaling limits of the ISC show interesting differences between low and high dimensions* » donne les syntagmes *the scaling limit of the ISC*, *the scaling limit*, *the ISC*, *interesting difference between low and high dimension*, *interesting difference*, *low and high dimension*, *high dimension*.

4.2. Evaluation

4.2.1. Du probabiliste au strict

Sachant que notre corpus attribue une catégorie unique à un document, nous avons décidé de retenir pour chaque document le thème avec le poids le plus élevé, transformant de fait la classification probabiliste de LDA en classification stricte.

La construction du modèle LDA nécessite la spécification du nombre de thèmes (k). Sachant que notre expérience se déroulera sur un corpus catégorisé, nous pouvons fournir k à LDA. Il y a donc k classes strictes, exactement autant que le nombre de catégories qui lui sont fournies ; ce qui nous permettra d'apparier un groupe identifié par une interprétation binaire du résultat de LDA à une et une seule catégorie du corpus.

S'agissant d'une expérience préliminaire, nous nous limiterons à cette configuration. Une évolution vers une appartenance probabiliste à plusieurs thèmes est à prévoir dans de futurs travaux.

4.2.2. Comparaison catégorie-groupe

Pour une collection donnée $\mathcal{D}$ pour laquelle nous disposons de ses k catégories et pour un résultat de LDA suivi de l'algorithme d'alignement, nous pouvons considérer deux partitions de $\mathcal{D}$:
i) Catégories existantes : $C_1, C_2, \ldots C_k$
ii) Groupes calculés : $G_1, G_2, \ldots G_k$
avec l'alignement $C_i \leftrightarrow G_i, i = 1, k$.

Idéalement nous devrions obtenir la parfaite superposition : $G_i = C_i$, mais le plus souvent les deux ensembles G_i et C_i sont différents. Afin de mesurer l'écart entre ces deux ensembles, nous reprenons les notions de précision, rappel et F_1-mesure.

4.2.3. Alignement

La F_1-mesure peut être interprétée comme une mesure de similarité entre le groupe G et la catégorie C. Le score de comparaison entre deux partitions alignées $C_1, C_2, \ldots C_k$ et $G_1, G_2, \ldots G_k$ est pris comme la moyenne des F_1-mesures :

$$Score((C_1, C_2, \ldots C_k) \leftrightarrow (G_1, G_2, \ldots G_k)) = \frac{\sum_{i=1}^{k} F_1(G_i, C_i)}{k}$$

L'alignement des groupes identifiés par LDA avec les catégories du corpus est basé sur la matrice de confusion qui contient les F_1-mesures présentées ci-dessus pour chaque couple catégorie-classe. L'objectif de cette étape d'alignement consiste à optimiser le score de l'ensemble des partitions, mesure qui nous permettra de comparer les performances entre les différentes exécutions.

Algorithm 1 Algorithme d'alignement

Input : `cats` les `k` catégories du corpus gold, `clusters` les `k` groupes calculés par LDA

Output : `result` un ensemble de `k` couples catégorie-groupe

```
1: cm : matrix[k][k]
2: for i in 1, k do
3:     for j in 1, k do
4:         cm[i][j] ← getF1Measure(clusters[i],cats[j])
5:     end for
6: end for
7: result ← ∅
8: while size(result) < k do
9:     for i in 1, size(clusters) do
10:        jmax ← getArgMax(cm[i][])
11:        if getMaxSim(cm[][jmax]) == cm[i][jmax] then
12:            result.add(<clusters[i], cats[jmax]>)
13:            clusters.remove(clusters[i])
14:            cats.remove(cats[jmax])
15:            cm[i][].remove()
16:            cm[][jmax].remove()
17:        end if
18:    end for
19: end while
```

La logique est la suivante :

1. Pour chaque groupe, repérer de quelle catégorie il est le plus proche (F_1-mesure maximum),

2. S'il n'y a pas d'autre groupe avec une similarité supérieure, l'association groupe-catégorie est validée et est ajoutée au résultat,

3. S'il reste des éléments à aligner (i.e. la taille de *result* est inférieure au nombre de catégories), alors itération.

Cette proposition est proche des l'algorithmes dits des mariages hongrois mais avec une complexité inférieure ($O(k^3)$ au lieu de de $O(k^4)$).

4.2.4. Jeu d'exécutions

L'algorithme de construction du modèle LDA que nous avons utilisé [1] n'est pas déterministe. Pour chaque collection avec un ensemble de catégories prédéfinies, nous avons réalisé 100 exécutions et nous avons gardé pour chaque exécution la valeur du score d'alignement $Score((C_1, C_2, \ldots C_k) \leftrightarrow (G_1, G_2, \ldots G_k))$.

5. Aperçu de l'expérience

5.1. Corpus

Nous avons sélectionné le corpus anglais ISEARCH (Lykke *et al.*, 2010), constitué de dizaines de milliers documents scientifiques et techniques catégorisés et hiérarchisés ; et plus spécifiquement ceux qui sont des résumés car seuls ceux-ci contiennent du texte directement exploitable.

Nous avons retenu ce corpus pour ses catégories très spécifiques qui nous laissent penser que l'exploitation des syntagmes aura un intérêt notable. Nous en avons retenu un extrait comme corpus d'évaluation pour nos expériences. Nous avons sélectionné les catégories et sous-catégories suivantes parmi celles proposées dans le corpus complet d'ISEARCH (les intitulés sont d'origine) :

Mathematics : Probability, Analysis of PDEs, Quantum Algebra, Differential Geometry, Algebraic Geometry.

Physics : General Physics, Optics, Atomic Physics, Chemical Physics, Fluid Dynamics.

High Energy Physics : Experiment, Lattice, Phenomenology, Theory.

Condensed Matter : Statistical Mechanics, Strongly Correlated Electrons, Materials Science, Superconductivity, Mesoscale and Nanoscale Physics

L'extrait comprend quatre grandes catégories et leurs sous-catégories respectives, et un échantillon de 750 à 1 000 documents par sous-catégorie selon les disponibilités dans le corpus. L'organisation des différentes exécutions de LDA suit l'énumération ci-dessus : pour *Mathematics*, nous commençons avec les deux catégories *Probability* et *Analysis of PDEs*, pour ajouter *Quantum Algebra*, puis *Differential Geometry* et enfin *Algebraic Geometry*. Les exécutions de LDA sur les autres grandes catégories suivent la même logique, à savoir l'ordre d'énumération ci-dessus. Ne disposant pas d'expert

1. https://github.com/chen0040/java-lda

pouvant spéculer sur des similarités potentielles entre sous-catégories, l'ordre établi est entièrement arbitraire.

Nous avons appliqué certaines contraintes aux documents du corpus. Nous avons supprimé tous les documents appartenant à plusieurs des catégories que nous avons retenues. Nous avons également imposé une limite de taille minimale (cent caractères) et au moins une phrase correcte syntaxiquement afin d'éviter d'introduire du bruit au travers de documents non valides. Ces deux contraintes ont éliminé 1 331 documents. Les mots vides/grammaticaux ont été supprimés.

5.2. Prétraitements

Quatre jeux d'exécutions sont à distinguer, chaque jeu représentant cent exécutions distinctes de LDA respectivement appliquées aux combinaisons évoquées ci-dessus :

1. LDA standard (unigrammes), à savoir la version sans prétraitement (LDA),
2. LDA après lemmatisation (LDA-L),
3. LDA après lemmatisation et extraction des syntagmes (LDA-LS),
4. LDA après lemmatisation, extraction des syntagmes et filtrage (LDA-LSF).

Par *filtrage* nous désignons la méthode de sélection des éléments retenus — à partir de leur étiquette morphosyntaxique (noms, verbes, adjectifs) (voir section 4.1). Comme évoqué plus haut (voir section 4.1), l'introduction de l'ACI se fait via la substitution/complétion du document original par un ensemble de syntagmes extraits par cette analyse. Le résultat de cette transformation par ACI n'est donc pas un texte humainement compréhensible mais une combinaison de chaînes de caractères prétraitées et éventuellement filtrées.

5.3. Outils & paramètres

Deux librairies ont été utilisées dans nos expériences : i) STANFORD CORENLP et ii) LDA. Pour STANFORD CORENLP, le seul paramétrage consiste à choisir les modèles de langue utilisés, à savoir ceux fournis avec la version 3.7.0 de la librairie [2].

L'implémentation de LDA que nous avons choisie requiert les informations suivantes :

1. le nombre de thèmes à identifier : le nombre de catégories injectées, de 2 à 5,
2. la taille maximum du vocabulaire (40 000),
3. le nombre d'itérations lors du traitement (fixée à 500 dans cette expérience),
4. une liste de mots vides (la liste de NLTK (Bird, Loper, 2004)),
5. deux variables α et β :

2. https://mvnrepository.com/artifact/edu.stanford.nlp/stanford-corenlp/3.7.0

α**:** Plus la valeur est petite, plus LDA tend à associer un document à peu de groupes ($\alpha = 0.1$ pour l'expérience).

β**:** Plus la valeur est petite, plus les vocabulaires des groupes sont distincts ($\beta = 0.01$).

La taille maximum du vocabulaire a été fixée relativement à nos observations sur le corpus. Un seuil à 40 000 permet de conserver tous les termes du vocabulaire qui apparaissent au moins deux fois dans le corpus tout en minimisant le nombre d'hapax. Les valeurs de α et de β ont été fixées empiriquement.

5.4. Résultats

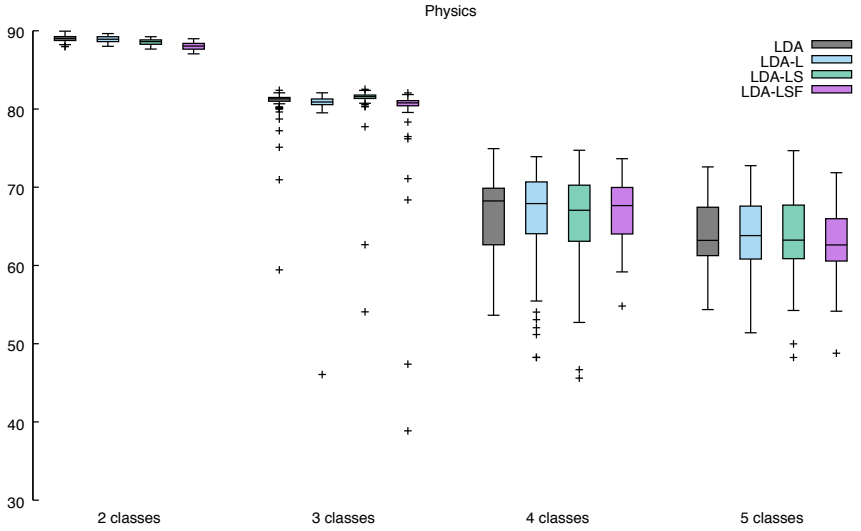

Figure 1. Représentations des scores obtenus pour la détection des sous-catégories de «Physics»

Nous avons traité les quatre grandes catégories comme quatre corpora indépendants. Les variations du score d'alignement pour ces catégories sont représentées synthétiquement dans les figures 1, 2, 3 et 4.

Sur les quatre figures et pour les corpus à deux classes, nous pouvons observer des F_1-mesures qui varient de 0,88 à 0,98. Ces valeurs très élevées montrent que la catégorisation du corpus est de bonne qualité, et que le choix de LDA comme modèle de thèmes est approprié.

Les figures 1 et 2 — respectivement *Physics* et *Condensed Matter* — illustrent une augmentation de la variation des résultats du modèle en fonction du nombre de classes (écart inter-quartile), mais elles ne présentent que peu de variations entre les

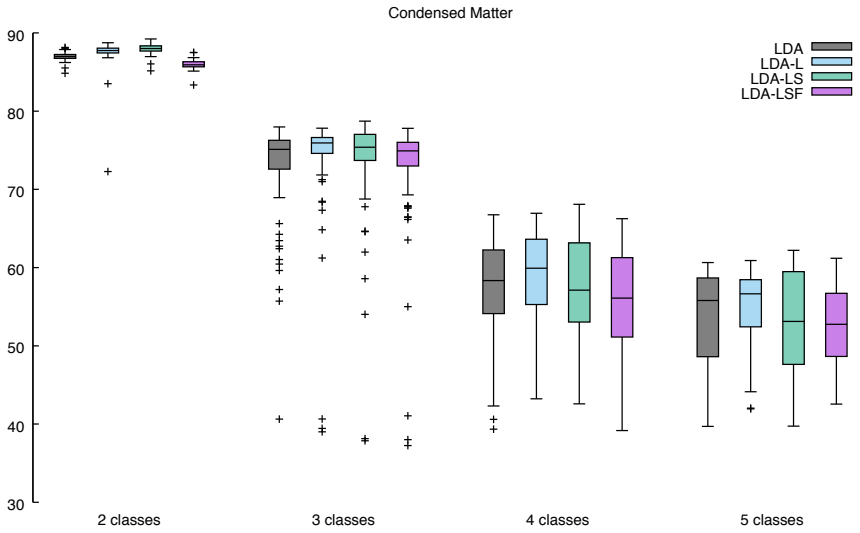

Figure 2. Représentations des scores obtenus pour la détection des sous-catégories de «Condensed Matter»

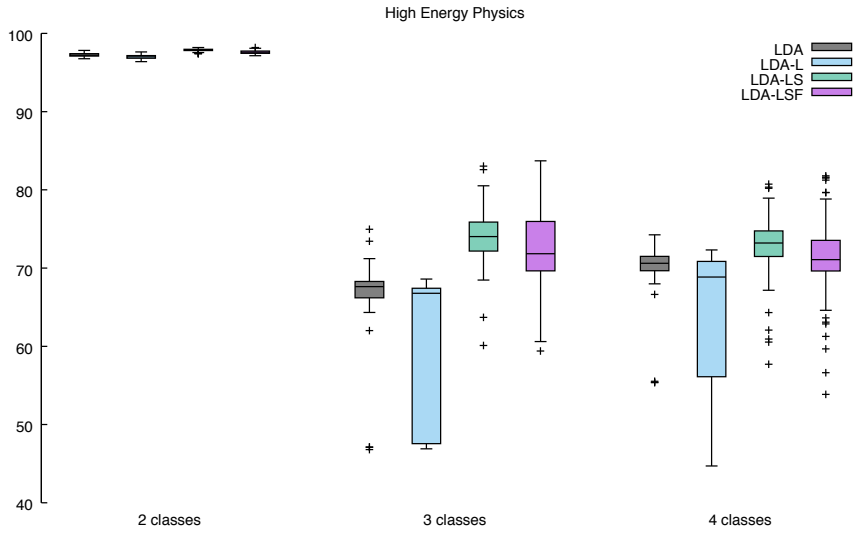

Figure 3. Représentations des scores obtenus pour la détection des sous-catégories de «High Energy Physics»

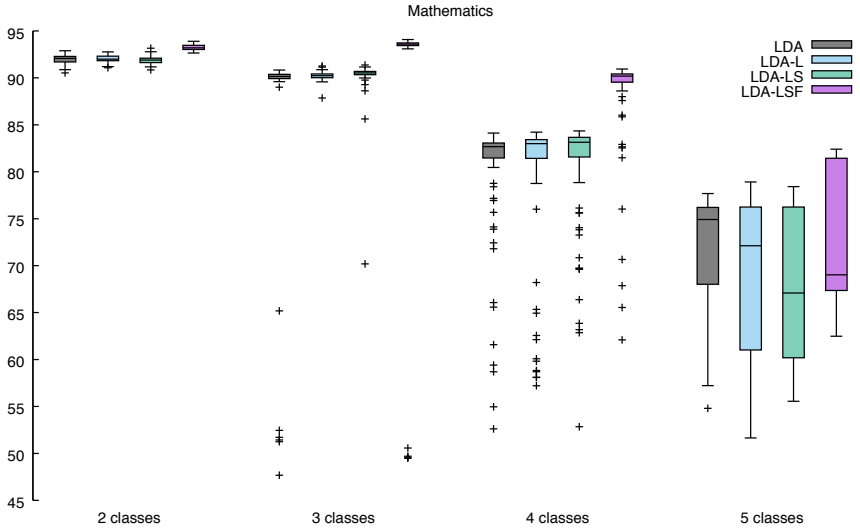

Figure 4. Représentations des scores obtenus pour la détection des sous-catégories de «Mathematics»

jeux d'exécutions — et ce indépendamment du nombre de classes. Cela peut s'expliquer par une analyse du vocabulaire exploité pour les différents jeux. Les données des jeux LDA-LS et LDA-LSF nous informent qu'en moyenne, pour ces jeux, 83% du vocabulaire est constitué d'hapax. En comparaison, les vocabulaires des jeux LDA et LDA-L comprennent 53% d'hapax. Cela peut s'expliquer par notre choix d'extraire le maximum de syntagmes, mais leur impact reste limité par la taille fixe du vocabulaire du modèle. Il est intéressant de noter que pour les corpora *Physics* et *Condensed Matter*, l'ajout de syntagmes n'apporte que peu voire pas de gain de performances. Nous ne pouvons cependant pas en tirer de conclusion générale dans la mesure où les résultats sur les corpora *Mathematics* et *High Energy Physics* sont divergents.

Contrairement aux figures 1 et 2, les figures 3 et 4 présentent d'importantes variations entre les jeux d'exécutions. Nous pouvons constater une perte d'une trentaine de points de F_1-mesure pour tous les jeux sur le corpus *High Energy Physics* (figure 3) lors de l'ajout de la troisième classe. Alors qu'à deux classes tous les scores de tous les jeux sont supérieurs à $0{,}95$, la troisième classe semble confondre le modèle en erreur. Une interprétation possible est que la distribution du vocabulaire de la troisième classe est similaire à la distribution du vocabulaire d'une classe déjà existante. Cette interprétation est étayée lors de l'ajout de la quatrième classe, qui ne provoque que de faibles baisses dans les jeux avec syntagmes et des améliorations significatives pour les deux autres jeux. Nous pouvons émettre l'hypothèse que la distribution du vocabulaire de la quatrième classe est particulièrement distincte des distributions des classes existantes, ce qui expliquerait une quasi stabilité voire une amélioration des résultats alors que l'inverse est attendu quand une classe supplémentaire est ajoutée. Hypothèse

vérifiée quand les ajouts des troisième et quatrième classes sont inversés : la chute de F_1-mesure est reportée à l'ajout de la dernière classe. Les résultats sur le corpus *High Energy Physics* semblent indiquer que l'exploitation des syntagmes est utile dans les cas où le vocabulaire des thèmes à identifier est particulièrement partagé.

C'est sur le corpus *Mathematics* (figure 4) que nous avons obtenu les meilleurs résultats en termes de qualité du modèle avec syntagmes. Aucune chute de la F_1-mesure aussi importante que précédemment ne peut être observée et il y a d'importantes variations entre les jeux d'exécutions. Nous pouvons observer que dans trois jeux sur quatre, le jeu LDA+LSF est significativement plus performant que les autres. Les différences de résultats doivent être mises en relation avec les vocabulaires respectifs de chaque catégorie : nous avons obtenu les meilleurs résultats avec les syntagmes sur les catégories avec les vocabulaires les plus restreints ($|\mathcal{V}_{\text{Mathematics}}| = 78\,507$ et $|\mathcal{V}_{\text{HighEnergyPhysics}}| = 104\,060$) et de moins significatifs sur les catégories avec de plus larges vocabulaires ($|\mathcal{V}_{\text{Physics}}| = 173\,957$ et $|\mathcal{V}_{\text{CondensedMatter}}| = 165\,352$).

6. Conclusions & perspectives

Le corpus d'expérimentation est découpé en quatre catégories (*Mathematics*, *Physics*, *Condensed Matter* et *High Energy Physics*) ; sur chacune nous avons tenté de distinguer leurs sous-catégories en exécutant LDA avec des prétraitements qui intègrent différents niveaux de traitement automatique de la langue.

Nous avons pu observer une quasi stabilité des résultats sur les catégories *Physics* et *Condensed Matter* quel que soit le niveau de prétraitement, et une augmentation sur les catégories *High Energy Physics* et *Mathematics* avec notre prétraitement le plus élaboré qui intègre la lemmatisation et la sélection de certains syntagmes (LDA-LSF). Cependant nous avons pu observer d'importantes variations des résultats de LDA en fonction du nombre de sous-catégories à identifier et de la catégorie analysée. Nos résultats semblent montrer un gain de performances lors de l'ajout des syntagmes pour les catégories à vocabulaire plus limité, ce qu'il nous faudra expérimenter davantage.

Il nous faudra également confirmer notre hypothèse que la similarité des distributions de vocabulaires induit des erreurs de classification liées à une mauvaise construction du modèle. Un travail de conflation des syntagmes permettra de diminuer le nombre d'hapax et d'augmenter le partage des vocabulaires entre les documents. Il serait intéressant d'observer le comportements des modèles qui descendent de LDA (Kim *et al.*, 2012 ; Paisley *et al.*, 2015) avec les mêmes prétraitements linguistiques et sur le même corpus d'expérimentation. Il faut cependant mettre en regard nos résultats et le corpus analysé, ce dernier étant de taille restreinte et très spécialisé. Pour plus de généralisation, il faudrait reproduire l'expérience en alternant les modèles de thèmes et les corpus. Des modèles de thèmes comme le Chinese Restaurant Process (CRP) permettraient d'obtenir directement une classification stricte.

Bibliographie

Andrzejewski D., Zhu X., Craven M., Recht B. (2011). A framework for incorporating general domain knowledge into Latent Dirichlet Allocation using first-order logic. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, p. 1171.

Arun R., Saradha R., Suresh V., Murty M., Madhavan C. (2009). Stopwords and stylometry: a Latent Dirichlet Allocation approach. In *NIPS workshop on Applications for Topic Models.*

Bird S., Loper E. (2004). NLTK: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, p. 31.

Bíró I., Siklósi D., Szabó J., Benczúr A. A. (2009). Linked Latent Dirichlet Allocation in web spam filtering. In *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*, p. 37–40.

Bíró I., Szabó J. (2009). Latent Dirichlet Allocation for automatic document categorization. In *Joint european conference on machine learning and knowledge discovery in databases*, p. 430–441.

Bíró I., Szabó J., Benczúr A. A. (2008). Latent Dirichlet Allocation in web spam filtering. In *Proceedings of the 4th international workshop on Adversarial information retrieval on the web*, p. 29–32.

Blei D. M., Lafferty J. D. (2009). Visualizing topics with multi-word expressions. *arXiv preprint arXiv:0907.1013*.

Blei D. M., Ng A. Y., Jordan M. I. (2003). Latent Dirichlet Allocation. *Journal of machine Learning research*, vol. 3, p. 993–1022.

Bloomfield L. (1933). *Language*. Holt, Rinehart and Winston.

Bolelli L., Ertekin Ş., Giles C. L. (2009). Topic and trend detection in text collections using Latent Dirichlet Allocation. In *European Conference on Information Retrieval*, p. 776–780.

Griffiths T. L., Steyvers M., Blei D. M., Tenenbaum J. B. (2004). Integrating topics and syntax. In *Advances in Neural Information Processing Systems*, p. 537–544.

Hofmann T. (1999). Probabilistic Latent semantic indexing. In *Proceedings of the 22nd annual international acm sigir conference on research and development in information retrieval*, p. 50–57.

Kim J. H., Kim D., Kim S., Oh A. (2012). Modeling topic hierarchies with the recursive chinese restaurant process. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, p. 783–792. ACM.

Liu Z., Li M., Liu Y., Ponraj M. (2011). Performance evaluation of Latent Dirichlet Allocation in text mining. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on*, vol. 4, p. 2695–2698.

Lykke M., Larsen B., Lund H., Ingwersen P. (2010). Developing a test collection for the evaluation of integrated search. In *European Conference on Information Retrieval*, p. 627–630.

Nokel M., Loukachevitch N. (2015). A method of accounting bigrams in topic models. In *Proceedings of the 11th workshop on multiword expressions*, p. 1–9.

Nokel M., Loukachevitch N. (2016). Accounting ngrams and multi-word terms can improve topic models. In *Proceedings of the 12th Workshop on Multiword Expressions*, p. 44–49.

Paisley J., Wang C., Blei D. M., Jordan M. I. (2015). Nested hierarchical Dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, n° 2, p. 256–270.

Phan X.-H., Nguyen L.-M., Horiguchi S. (2008). Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*, p. 91–100.

Qian S., Zhang T., Xu C., Hossain M. S. (2015). Social event classification via boosted multimodal supervised Latent Dirichlet Allocation. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 11, n° 2, p. 27.

Tian K., Revelle M., Poshyvanyk D. (2009). Using Latent Dirichlet Allocation for automatic categorization of software. In *Mining Software Repositories, 2009. MSR'09. 6th IEEE International Working Conference on*, p. 163–166.

Vapnik V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Wallach H. M. (2006). Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on machine learning*, p. 977–984.

Wang X., McCallum A., Wei X. (2007). Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, p. 697–702.