Les itemsets partagés au service de la classification textuelle

Louis Rompré[1], Ayoub Bokhabrine[2], Ismaïl Biskri[3] et Nadia Ghazzali[4]

[1]Université du Québec à Trois-Rivières – Louis.Rompre@uqtr.ca

[2] Université du Québec à Trois-Rivières– Ayoub.Bokhabrine@uqtr.ca

[3] Université du Québec à Trois-Rivières – Ismail.Biskri@uqtr.ca

[4] Université du Québec à Trois-Rivières – Nadia.Ghazzali@uqtr.ca

Abstract

Finding and evaluating descriptors which allow to create classes of similarities is the foundation of all automated classification processes. Choosing a descriptor instead of another one may have a huge impact on result. The distribution of descriptors within the documents is a clue used by the classifiers to determine the classes of similarity to which the documents belong. This article proposes a comparative study of three descriptors used in automatic text processing: n-grams of characters, words and itemsets. The level of abstraction of these descriptors varies greatly. They give the existing relationships between the words that appear together in a text. Compare two words or n-grams, the semantic contribution of this descriptor is more significant. We propose a method and the addition of a constraint to identify the relevant itemsets distributed in textual documents. Experiments conducted demonstrate that the classes produced using the proposed method are better than those produced with words or n-grams.

Keywords: Descriptor, Clustering, Class, Text, Word, N-Gram, Itemset, K-Means.

Résumé

L'identification et l'évaluation de descripteurs qui permettent de distinguer une classe de similarité d'une autre est à la base du processus de classification. Le choix d'un descripteur plutôt qu'un autre a un impact majeur sur la qualité des résultats obtenus. Il influence le comportement d'un classifieur, sa présence ou non étant un indice permettant de cibler la classe à laquelle appartient un document. Cet article propose une étude comparative de trois descripteurs utilisés en traitement automatique du texte et dont le niveau d'abstraction varie, soit les n-grammes de caractères, les mots et les itemsets. Les itemsets représentent les relations de cooccurrence qui existent entre des mots qui composent un texte. Par rapport aux mots ou aux n-grammes, l'apport sémantique de ce descripteur est plus significatif. Nous proposons une méthode et l'ajout d'une contrainte pour dégager des itemsets pertinents répartis dans des documents textuels. Les expérimentations effectuées suggèrent que les classes produites à l'aide de la méthode proposée sont de meilleure qualité que celles produites avec les mots ou les n-grammes.

Mots clés : Descripteur, Clustering, Classe, Texte, Mot, N-Gramme, Itemset, K-means.

1. Introduction

Les percées dans le domaine de l'apprentissage machine et, en particulier, en apprentissage profond (deep learning) ont accéléré la recherche dédiée à l'automatisation du processus de

classification d'images (LeCun et al., 2015), de sons (Hinton, 2012) et de documents textuels (Zhang et al., 2015; Lai et al., 2015). Bien que l'apprentissage profond soit omniprésent, d'autres méthodes telle que, par exemple, la classification naïve bayésienne (Xu, 2018) ou SVM (Goudjil et al., 2018) demeurent étudiées. Plus souvent qu'autrement, l'attention est portée sur la sélection et la configuration des classifieurs. Plusieurs études publiées à travers les années démontrent comment différentes configurations peuvent impacter positivement ou négativement les classifications produites (Hartmann et al., 2019; Joulin et al., 2016; Aggarwal et Zhai, 2012). Bien que l'évaluation du comportement des classifieurs soit d'intérêt, mesurer l'influence des descripteurs utilisés comme intrant demeure primordial car les classifieurs n'opèrent pas directement sur les documents textuels mais plutôt sur des représentations simplifiées de ceux-ci. Les documents sont transposés en vecteurs contenant différents descripteurs à partir d'une opération qui consiste à cibler et comptabiliser des informations jugées pertinentes et réparties à l'intérieur des documents. La présence ou l'absence de descripteurs spécifiques déteint sur le comportement des classifieurs. Les descripteurs sélectionnés doivent être suffisamment discriminants pour permettre aux classifieurs de générer des classes de similarités cohérentes.

La sélection des descripteurs à utiliser pour automatiser le processus de classification est une discipline qui, encore aujourd'hui, repose sur l'intuition et l'expérience. Nous proposons une méthode qui exploite la notion d'itemsets partagés. Cette méthode garantit l'achèvement des traitements dans un temps raisonnable tout en produisant des résultats meilleurs que ceux obtenus lorsque les mots ou les n-grammes de caractères sont utilisés.

2. La classification textuelle automatisée

Les différentes méthodes de classification automatisée se distinguent par les algorithmes qui les composent. Elles possèdent toutes des avantages et des limitations. Par exemple, les réseaux de neurones profonds génèrent des résultats intéressants et ce même avec de vastes ensembles de données. Toutefois, les réseaux de neurones profonds demeurent difficiles à interpréter. Ce qui est problématique lorsqu'il est nécessaire d'expliquer la pertinence des résultats (Kowsari et al., 2019). Au-delà des algorithmes, la performance des méthodes de classification automatisée dépend fortement de la nature, la quantité, la qualité ou encore la fiabilité des données sur lesquelles elles opèrent. La performance dépend également des objectifs visés et des connaissances du sujet abordé. Sans une connaissance adéquate du domaine une interprétation et une évaluation erronées des résultats est possible. L'implication d'un expert du domaine demeure donc essentielle.

Il est généralement admis que la qualité d'une classification est déterminée en fonction de l'homogénéité de ses classes. Une classification est jugée de qualité lorsque, d'une part, les membres d'une même classe se ressemblent et, d'autre part, que les membres des différentes classes diffèrent. Il revient donc aux méthodes de classification automatisée de cibler les informations pertinentes qui permettent le bon partitionnement des données. Les particularités des algorithmes de classification supervisée et non supervisée, et leur application pour le traitement du texte sont couverts dans la littérature (Xu et Tian, 2015; Berkhin, 2006).

3. Les descripteurs

Pour la classification textuelle, le mot et le n-gramme de mot ou de caractères ont, longtemps, été utilisés comme descripteurs discriminants.

L'usage des mots pour décrire un texte est pratiquement une norme et la fréquence d'apparition de ces mots dans un document textuel est fréquemment utilisée pour les caractériser. L'utilisation de ces fréquences d'apparition des mots s'accompagne, souvent, de l'élimination des mots vides et de lemmatisation. Ces prétraitements améliorent, généralement, la qualité des résultats produits. Les mots vides ne sont pas informatifs. La plupart du temps, ils sont largement distribués dans les textes. Éliminer les mots vides réduit le risque que des liens non pertinents soient établis entre des textes abordant des sujets différents lorsque la fréquence d'apparition des mots est utilisée comme descripteur. Inversement, ramener les mots à leur forme la plus simple permet d'augmenter la fréquence d'apparition de mots informatifs.

Un n-gramme de caractères est une succession de *n* caractères consécutifs. Par exemple, le mot « texte » est formé des tri-grammes suivants : « tex », « ext », et « xte ». Bien que l'apport sémantique de ce descripteur soit faible, le n-gramme de caractères est réputé pour être tolérant aux fautes d'orthographe et aux fautes grammaticales.

Quand plusieurs mots ou n-grammes apparaissent à des fréquences comparables et régulières dans deux documents ou portions de documents, ces documents sont considérés similaires. Néanmoins, il est courant que des documents, traitant de thèmes distincts, partagent un certain nombre de n-grammes ou de mots qui n'ont pas la même acception sémantique. Ces documents sont sujets à être associés à différentes classes de similarité par les utilisateurs qui les consultent. Par conséquent, les mots ou les n-grammes seuls deviennent peu informatifs et ne peuvent supporter adéquatement le processus classificatoire. Ils peuvent même mener à l'introduction de fausses indications. Leur utilité pour établir le degré de similarité entre des documents est, dès lors, limitée voire non pertinente.

3.1. Les itemsets

Les relations de cooccurrence entre certains mots issus d'un même segment de texte peuvent servir à préciser la sémantique de ce segment. Dans ce contexte, un segment peut être, un chapitre, une phrase, un paragraphe ou un texte complet. Ces relations de cooccurrence peuvent être exprimées à l'aide d'itemsets. L'essor de ce descripteur découle, principalement, des travaux d'Agrawal (Agrawal et al., 1993). De manière générale, on nomme itemsets tous les sous-ensembles qu'il est possible de générer à partir d'un ensemble d'items distincts. Par exemple, à partir des items i_1, i_2 et i_3 il est possible de générer les itemsets « i_1 », « i_2 », « i_3 », « i_1, i_2 », « i_1, i_3 », « i_2, i_3 » et « i_1, i_2, i_3 ». La notion d'item est définie en fonction du domaine abordé. Pour le texte, les mots sont naturellement utilisés (Rompré et Biskri, 2018; Bokhabrine, Biskri et Ghazzali, 2019). Un itemset est considéré fréquent lorsque sa fréquence d'apparition est supérieure à un seuil prédéterminé. Les itemsets fréquents peuvent être dégagés à l'aide des algorithmes Apriori (Agrawal et Srikant, 1994) ou FP-Growth (Han et al., 2000).

Pour un ensemble de taille *d*, le nombre d'itemsets possible est 2^d (Tan et al., 2002) ou 2^d-1 excluant l'ensemble vide. Le vocabulaire étant enrichi continuellement, il est difficile de recenser avec exactitude le nombre de mots que renferme une langue comme le français. En admettant que ce nombre est supérieur à 30 000, il est possible de générer plus de $2^{30\,000}$ itemsets à partir du lexique français. Heureusement, ce n'est pas la totalité du lexique qui est examinée mais plutôt les mots distincts contenus dans les textes. Le nombre de mots distincts contenus dans un texte dépend de plusieurs facteurs dont la longueur et le style d'écriture. Néanmoins, l'utilisation des itemsets comme descripteurs de documents textuels requiert des

mécanismes pour contrôler le nombre d'items à considérer afin d'éviter une explosion combinatoire.

4. Méthodologie

Nous proposons d'utiliser des « itemsets partagés » pour grouper automatiquement des documents textuels.

L'utilisation des itemsets fréquents est souvent problématique. Ils peuvent être trop génériques ou être trop nombreux pour pouvoir être traités (Holat et al., 2015). Aussi, nous proposons deux mécanismes pour rehausser leur utilité et ainsi soutenir efficacement la classification automatisée de documents textuels.

Le premier est l'ajout d'une étape de filtrage avant la génération des itemsets, qui permet d'éliminer tous les mots du document qui sont peu fréquents. L'algorithme Apriori intègre un mécanisme similaire. Toutefois, lorsqu'il recherche les itemsets fréquents, l'algorithme génère et évalue tous les itemsets qu'il est possible de créer à partir des items fréquents. Cette étape est extrêmement coûteuse et considérée contreproductive puisque de nombreux itemsets évalués n'apparaissent pas dans le texte. Pour cette raison, l'algorithme FP-Growth est préféré. Cet algorithme exploite une structure de données en forme d'arbre qui lui permet d'éviter de générer et d'évaluer tous les candidats possibles. Même avec cette amélioration, le temps d'exécution et la mémoire utilisée pour dégager les itemsets fréquents demeurent critique lorsque le nombre d'items est important. L'ajout d'une étape de filtrage en amont permet de contenir le nombre d'items à traiter.

Le second mécanisme est l'introduction d'une contrainte qui limite le nombre d'itemsets fréquents considérés pour regrouper automatiquement les documents. Un itemset peut à la fois être fréquent pour un document et absent pour d'autres. Ainsi, les itemsets qui apparaissent uniquement dans un seul document sont écartés. L'hypothèse soutenue est que bien qu'ils puissent servir à décrire plus précisément le contenu des documents auxquels ils réfèrent, leur utilité pour créer des liens entre les documents est limitée et par conséquent ils peuvent être écartés.

La méthode proposée est composée de 5 étapes :

1) **Nettoyer les données.** Les mots vides sont éliminés. Les chiffres et certains caractères spéciaux sont également retirés. Cette opération permet de réduire le nombre de mots. Ce qui permet de restreindre l'espace de recherche lors de l'extraction des itemsets fréquents.
2) **Comptabiliser et filtrer les mots**. Pour chacun des documents, les mots sont comptabilisés et ceux dont la fréquence d'apparition est inférieure à un seuil $s_{\min}$ prédéterminé sont écartés.
3) **Extraire les itemsets fréquents**. L'algorithme FP-Growth est utilisé. Si le nombre d'itemsets fréquents retournés est inférieur à un nombre prédéterminé, $s_{\min}$ est diminué de 1, l'étape 2 est réexécutée et l'algorithme FG-Growth est appliqué de nouveau. Cette boucle se termine lorsque le nombre d'itemsets retournés est supérieur au nombre minimum fixé ou que $s_{\min} = 1$.
4) **Identifier les itemsets partagés**. Les itemsets fréquents de tous les documents sont comparés. Seuls ceux qui apparaissent dans plus d'un document sont conservés. Les itemsets partagés sont alors utilisés comme descripteurs.

5) **Grouper les documents similaires**. L'algorithme K-means est utilisé comme classifieur. Bien que cet algorithme ait été introduit il y a plusieurs années, il demeure largement employé pour la classification de documents textuels (Allahyari et al., 2017).

Des expérimentations ont été effectuées afin, d'une part de mesurer les effets de l'ajout du filtrage de mots en amont de l'étape d'extraction des itemsets fréquents et, d'autre part, de comparer l'apport des itemsets par rapport à celui des mots et des n-grammes pour regrouper automatiquement des documents textuels.

5. Expérimentations

Nous avons développé une application en Python pour évaluer la méthode proposée. Pour ce faire, les librairies mlxtend (Raschka, 2018) et Scikit-Learn (Pedregosa et al., 2011) ont été utilisées pour leur implémentation des algorithmes FP-Growth et K-Means respectivement. La méthode proposée a été appliquée à un ensemble de données composés de 48 documents textuels traitant de seize (16) sujets d'actualité (la légalisation du cannabis au Canada, la construction du pont Samuel-de-Champlain à Montréal, la crise du coronavirus, l'exclusion des athlètes russes des jeux olympiques de 2018, la mort du chanteur Johnny Hallyday, les ravages causés par les feux de forêt, les critiques du film Star Wars, l'imposition de Tesla, les records de vente de Boeing en 2017, Hyperloop, Brian Adams, la procédure de destitution de Donald Trump, la crise du recyclage au Québec, la fermeture d'une usine de Mega Brands à Montréal, la dépendance face au jeu vidéo Fortnite et le cours du bitcoin). Bien que les sujets abordés soient distincts, certains documents qui traitent de sujets différents contiennent des références aux mêmes concepts, lieux ou personnalités. Des références à la musique (concept) existent à la fois dans les documents traitant de la mort du chanteur Johnny Halliday et dans ceux traitant du chanteur Brian Adams. Des références à la ville de Montréal (lieu) se trouvent dans les documents traitant de la construction du pont Samuel-de-Champlain et dans ceux traitant de la fermeture d'une usine. Finalement, les documents dont le sujet est Tesla et ceux dont le sujet est l'Hyperloop contiennent des références à Elon Musk (personnalité).

5.1. Évaluation de l'ajout du filtrage des mots

Le temps de calculs qui est nécessaire pour extraire les itemsets fréquents est un facteur limitatif. Ce temps peut dépasser 24h pour traiter un seul document. Réduire ce délai est donc crucial pour valoriser l'usage des itemsets fréquents comme descripteurs. La durée des traitements est exponentielle en fonction du nombre de mots. Nous avons donc mesuré le temps nécessaire à l'extraction des itemsets fréquents avec et sans l'ajout du filtrage des mots en amont de l'extraction des itemsets fréquents.

Lors de cette expérimentation nous avons utilisé les plus petits textes de notre ensemble de données. Le texte 1 contient un total de 1032 mots. Après le nettoyage, il reste 15 mots distincts. Le texte 2 contient 1280 mots et possède 14 mots distincts. Finalement, le texte 3 contient 1293 mots et possède 22 mots distincts. Lorsque le filtrage est appliqué, le nombre de mots distincts des textes 1 et 2 est réduit à 5 tandis que le nombre de mots distincts du texte 3 est réduit à 6. La taille maximale des itemsets a été fixée à 4 pour restreindre leur nombre.

Le tableau 1 donne les temps d'exécution obtenus avec et sans filtrage. Il y est démontré une amélioration notable. Pour les textes 1 et 2, le temps d'exécution passe plus ou moins d'une demi-heure à quelques millisecondes. Le temps pour traiter le texte 3 passe du 1h15 sans filtrage à 3 millisecondes avec filtrage. Bien que les temps capturés dépendent de la puissance

de l'ordinateur utilisé pour effectuer les traitements, ils permettent, néanmoins de comparer la performance des deux options. L'amélioration constatée découle directement de la diminution du nombre de mots distincts considérés. Il est admis que la réduction du nombre de mots entraîne une perte d'information. Toutefois, notre hypothèse est que cette perte d'information sera peu significative dans le contexte de la classification de documents textuels.

	Temps **sans** filtrage	Temps **avec** filtrage
Texte 1	0:34:11.6045	0:00:00.0279
Texte 2	0:34:50.7154	0:00:00.0189
Texte 3	1:15:23.8714	0:00:00.0309

Tableau 1 : Temps d'exécution pour l'extraction des itemsets fréquents

5.2. Évaluation de la valeur des itemsets partagés comme descripteur

Deux expérimentations ont été menées pour mesurer la valeur des itemsets partagés comme descripteurs. Les classes produites à l'aide des itemsets partagés ont été comparées à celles produites lorsque les mots et les n-grammes sont employés. Une première expérimentation a été réalisée sans la contrainte rejetant les itemsets fréquents non distribués dans plus d'un document. Une deuxième expérimentation a été effectuée avec la contrainte. Lors des deux expérimentations, le filtrage des mots a été appliqué avant de lancer l'algorithme FP-Growth.

Les figures 1, 2 et 3 illustrent les résultats obtenus sans la contrainte. La zone grisée à l'intérieure des graphiques représente le pourcentage de classes considérées comme étant bruitées tandis que la partie en noir représente le pourcentage de classes jugées de bonne qualité. Une classe est dite de bonne qualité lorsqu'elle regroupe tous les textes qui traitent du même sujet et aucun autre. On remarque que l'algorithme K-Means performe mieux lorsque les mots ou les n-grammes sont utilisés comme descripteurs. Ce comportement peut s'expliquer par le fait que la plupart des itemsets fréquents extraits d'un texte sont propres à ce texte. En fonction des paramètres établis, les mots produisent les résultats les plus intéressants. Néanmoins, seulement 44 % des classes sont considérées comme étant de bonne qualité.

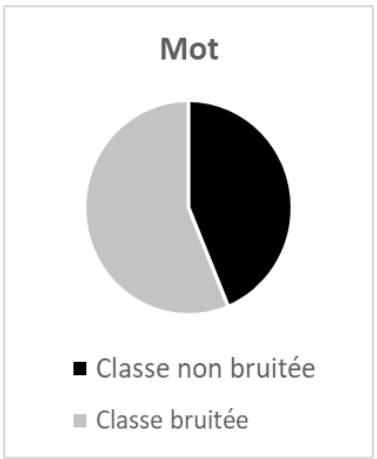

Figure 1 : Évaluation des classes obtenues avec les mots

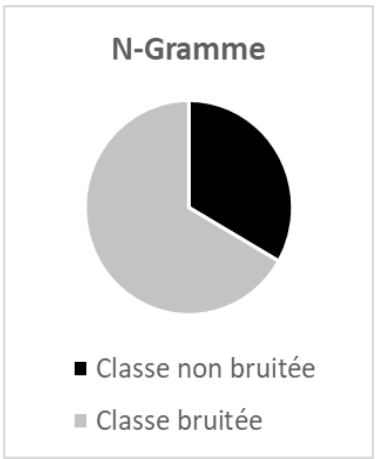

Figure 2 : Évaluation des classes obtenues avec les n-grammes

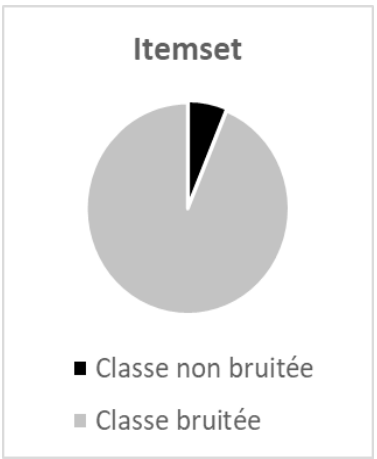

Figure 3 : Évaluation des classes obtenues avec les itemsets

Des travaux antérieurs ont démontré que les itemsets fréquents ont la capacité de décrire efficacement le contenu de documents textuels en plus de permettre d'établir des liens entre ces documents (Rompré et Biskri, 2018 ; Bokhabrine et al., 2019). Pour ce faire, un nombre restreint d'itemsets doit être exploité. Bien que l'absence d'une caractéristique puisse être un indice permettant d'identifier la classe de similarité à laquelle appartient un objet, la présence de caractéristiques partagées est d'autant plus utile. Ainsi, l'application de la contrainte qui veut qu'un itemset fréquent soit contenu dans plus d'un texte pour être retenu comme descripteur permet de réduire le nombre d'itemsets tout en conservant ceux susceptibles de favoriser la découverte de liens pertinents entre les documents.

Les figures 4, 5 et 6 illustrent les résultats recueillis lorsque la contrainte est appliquée. Ces résultats démontrent que le fait d'exploiter des itemsets partagés rehausse grandement la cohérence des classes de similarités produites. 88 % des classes générées à partir des itemsets partagés sont jugées de qualité. C'est le pourcentage le plus élevé. Ce résultat contraste avec celui obtenu lorsque la contrainte n'est pas appliquée.

L'ajout de la contrainte semble également être favorable aux mots. Lorsqu'elle est appliquée, le pourcentage de classes bruitées et produites à l'aide des mots diminue légèrement. Les résultats obtenus à partir des n-grammes sont toutefois peu ou pas influencés par l'ajout de la contrainte. En effet, comme l'illustrent les figurent 2 et 5, la proportion de classes bruitées demeure inchangée malgré l'ajout de la contrainte. La nature des n-grammes peut expliquer ce comportement. Les n-grammes sont massivement réparties dans les différents documents.

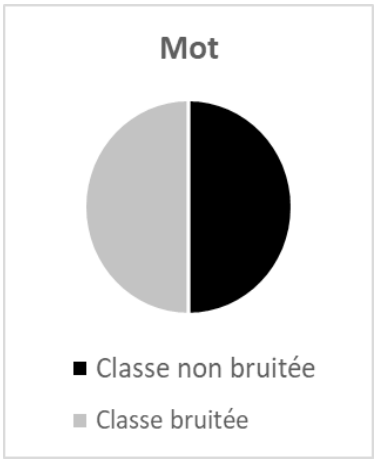

Figure 4 : Évaluation des classes obtenues avec les mots partagés

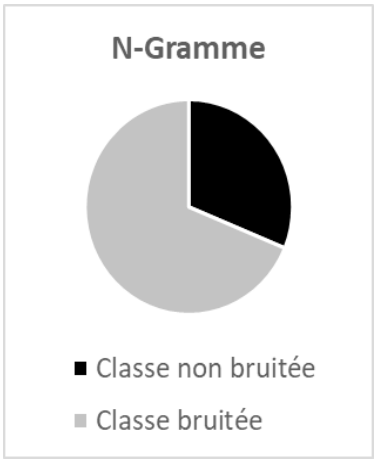

Figure 5 : Évaluation des classes obtenues avec les n-grammes partagés

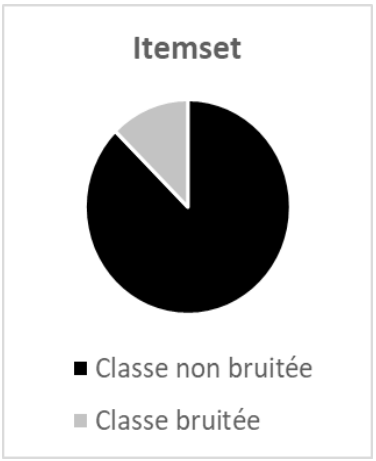

Figure 6 : Évaluation des classes obtenues avec les itemsets partagés

Les résultats obtenus suggèrent que les itemsets partagés sont plus discriminants que les mots seuls. Les itemsets partagés semblent séparer plus efficacement les documents qui partagent plusieurs références (concept, lieu, personnage) mais qui traitent de sujets différents. Ce comportement peut s'avérer utile lorsqu'il y a un recouvrement thématique entre les documents.

6. Conclusion

Nous avons proposé d'utiliser la notion d'itemsets partagés comme descripteurs. Les itemsets partagés représentent des groupes de mots qui cooccurrent fréquemment au sein des segments textuels qui composent des textes. Les expérimentations effectuées démontrent que les

itemsets partagés, même s'ils représentent une faible proportion des itemsets fréquents contenus dans un texte, peuvent être utilisés efficacement comme descripteurs. Bien que des informations soient « négligées » à la suite de la suppression de plusieurs itemsets non partagés, la méthode proposée permet la création d'une description compacte qui demeure très significative. Cette description peut servir à soutenir la classification automatisée des documents textuels.

Enfin, la méthode proposée permet de grouper automatiquement des documents textuels à l'aide d'itemsets partagés dans un temps raisonnable.

References

Aggarwal, C. C., & Zhai, C. (2012). A survey of text classification algorithms. In Mining text data (pp. 163-222). Springer, Boston, MA.

Agrawal, R., Imielinski T., et Swami, A. (1993). Minning association rules between sets of items in large databases, In Proc. of the SIGMOD Conference on Management of Data, pp 207-216.

Agrawal, R., Srikant, R. (1994). Fast Algorithms for Mining Association Rules, In Proc. of the 20th International Conference on Very Large Database, pp. 487-499

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). A brief survey of text mining: Classification, clustering and extraction techniques. arXiv preprint arXiv:1707.02919.

Berkhin, P. (2006). A survey of clustering data mining techniques. In Grouping multidimensional data (pp. 25-71). Springer, Berlin, Heidelberg.

Bokhabrine, A., Biskri, I., Ghazzali, N. (2019). Frequent Itemsets as Descriptors of Textual Records. In Proc. of the International conference ACM-ICCCI 2019, LNAI, Springer, pp 35-45.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. Information, 10(4), 150.

Goudjil, M., Koudil, M., Bedda, M., & Ghoggali, N. (2018). A novel active learning method using SVM for text classification. International Journal of Automation and Computing, 15(3), 290-298.

Hartmann, J., Huppertz, J., Schamp, C., & Heitmann, M. (2019). Comparing automated text classification methods. International Journal of Research in Marketing, 36(1), 20-38.

Han, J., Pei, J., et Yin, Y. (2000). Mining frequent patterns without candidate generation. In ACM sigmod record (Vol. 29, No. 2, pp. 1-12). ACM.

Holat, P., Tomeh, N., et Charnois, T. (2015). Classification de texte enrichie à l'aide de motifs séquentiels. TALN 2015.

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent convolutional neural networks for text classification. In Twenty-ninth AAAI conference on artificial intelligence.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.

MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 14, pp. 281-297).

Raschka, S. (2018). MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack. Journal of open source software, 3(24), 638.

Rompré, L. et Biskri, I. (2018). Les « itemsets fréquents » comme descripteurs de documents textuels. In Proc. JADT 2018.

Tan, P. N., Kumar, V., et Srivastava, J. (2002). Selecting the right interestingness measure for association patterns. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 32-41). ACM.

Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. Annals of Data Science, 2(2), 165-193.

Xu, S. (2018). Bayesian Naïve Bayes classifiers to text classification. Journal of Information Science, 44(1), 48-59.

Zaïane, O. R., et Antonie, M. L. (2002). Classifying text documents by associating terms with text categories. In Australian computer Science communications (Vol. 24, No. 2, pp. 215-222). Australian Computer Society, Inc.

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In Advances in neural information processing systems (pp. 649-657).