Agrégation de sac-de-sacs-de-mots pour la recherche d'information par modèles vectoriels

Vincent Claveau

IRISA – CNRS
Campus de Beaulieu, F-35042 Rennes
vincent.claveau@irisa.fr

Résumé. Cet article étudie l'intérêt de représenter les documents textuels non plus comme des sacs-de-mots, mais comme des sacs-de-sacs-de-mots. Au cœur de l'utilisation de cette représentation, le calcul de similarité entre deux objets nécessite alors d'agréger toutes les similarités entre sacs de chacun des objets. Nous évaluons cette représentation dans un cadre de recherche d'information, et étudions les propriétés attendues de ces fonctions d'agrégation. Les expériences rapportées montrent l'intérêt de cette représentation lorsque les opérateurs d'agrégation respectent certaines propriétés, avec des gains très importants par rapport aux représentations standard.

1 Introduction

La représentation sac-de-mots des documents (abrégée ici en BoW, *Bag-of-Words*) est très largement utilisée en recherche d'information (RI) et en traitement automatique des langues (TAL). Elle permet d'associer à un texte un descripteur unique basé sur l'ensemble des mots-formes qu'il contient. Cependant, cette représentation est parfois trop grossière pour certaines tâches. Plusieurs représentations alternatives ont été imaginées selon les cas et les informations disponibles. Les similarités entre objets complexes (graphes, arbres...) ont été extensivement étudiés (Bunke, 2000, *inter alia*), mais sont rarement utilisées en RI à cause de leur coût calculatoire. C'est pourquoi, dans beaucoup de cas, les travaux gardent une structure de données identique à celle des sacs-de-mots (même si ce sont des morphèmes, des n-grammes ou des syntagmes et non plus des mots qui sont manipulés). Dans cet article, nous nous intéressons à une extension simple de la représentation classique en sac-de-mots dans laquelle un objet est décrit par un multiensemble de sac-de-mots. Cette représentation en sac-de-sacs-de-mots (BoBoW, *Bag-of-Bags-of-Words* garde certaines propriétés calculatoires des BoW, mais nécessite de savoir comment agréger les résultats obtenus entre sacs-de-sacs. Dans son travail séminal en RI, Wilkinson (1994) l'utilise pour comparer une requête aux différentes portions d'un document et combiner les résultats, soit sur les similarités soit sur les rangs. Mais les quelques fonctions d'agrégation testées obtiennent des résultats inférieurs à un système vectoriel classique. En revanche, cette représentation a été utilisée avec succès dans des cadres particuliers en TAL (Ebadat et al., 2012) et en image (Kondor et Jebara, 2003). Elle est aussi à rapprocher des travaux sur la recherche d'information structurée (Luk et al., 2002) (la prise en compte du

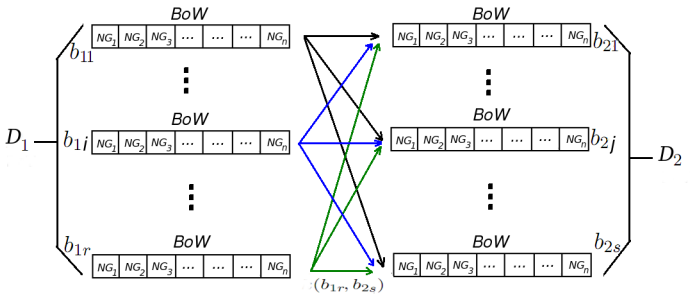

FIG. 1: Comparaison de deux sacs-de-sacs-de-mots basée sur les similarités des sacs-de-mots deux à deux.

contexte/structure pour indexer les éléments de ces documents est à rapprocher des propositions en sec. 3.2). La question cruciale, que nous explorons dans cet article, est d'étudier les propriétés des calculs de similarités entre deux sac-de-sacs-de-mots.

2 Sac de sacs de mots

2.1 Principe général

La représentation BoBoW consiste simplement à décrire un document comme un ensemble de sous-documents, chacun étant représenté par un sac-de-mots. Tout comme le sac-de-mots ne conserve pas l'ordre des mots, le sac-de-sacs ne conserve pas l'ordre des sous-documents. La comparaison de deux documents D_1 et D_2, ou d'une requête et d'un document, se fait en comparant chacun des BoW selon le modèle adopté et en en tirant une mesure globale de proximité entre D_1 et D_2. Dans la suite de cet article on note δ la mesure de comparaison définie entre deux BoW selon le modèle choisi. On considère dans la suite qu'il s'agit d'une mesure de similarité (e.g. un cosinus), mais les notions se transposent bien sûr quand δ est une dissimilarité (e.g. une distance L2).

Les documents peuvent être découpés en sous-documents de différentes façons, et le nombre de sous-documents par document peut être variable. Dans le cas du modèle vectoriel, chaque sous-document est un vecteur. Deux documents D_1 et D_2 sont notés ainsi : $D_1 = \{b_{1,1}, b_{1,2} \dots b_{1,i} \dots b_{1,r}\}$ et $D_2 = \{b_{2,1}, b_{2,2} \dots b_{2,j} \dots b_{2,s}\}$. La comparaison de deux documents nécessite le calcul de toutes les combinaisons de similarités entre les vecteurs des deux documents, comme illustré en figure 1.

Il faut noter que la représentation en sacs-de-sacs implique une plus grande complexité mémoire et calculatoire. Il faut en effet stocker plusieurs sous-documents pour un document ; ces sous-documents contiennent certes moins de mots, mais la somme de leur empreinte mémoire est supérieure à celle du document considéré dans son ensemble. Cette consommation supplémentaire dépend en pratique du nombre moyen de sacs par documents et de la façon dont ils sont stockés. En ce qui concerne le temps de calcul, la complexité est aussi plus grande. Au

moment de la recherche, le calcul exhaustif du score d'un document est en $\mathcal{O}(n * m * d)$ avec n le nombre de sacs de la requête, m celui du document et d la complexité du calcul de δ.

2.2 Propriétés des fonctions d'agrégation pour le modèle vectoriel

Les mesures similarités entre deux sacs-de-sacs-de-mots agrègent les résultats de la similarité mineure δ pour toutes les combinaisons possibles vecteur à vecteur. Deux façons simples pour ce faire ont été proposées (Haussler, 1999) :
Sum-Sum$(D, Q) = \sum_i \sum_j \delta(b_{Q,i}, b_{D,j})$ et Max-Max$(D, Q) = \max_i \max_j \delta(b_{Q,i}, b_{D,j})$
Plus récemment, une mesure plus générale a été proposée (Gosselin et al., 2007) :
PowerScalar$(D, Q) = \sqrt[q]{\sum_i \sum_j \delta(b_{Q,i}, b_{D,j})^q}$.
Le paramètre $q \in [0, \infty[$ donne plus ou moins d'importance aux valeurs hautes et contrôle donc le pouvoir discriminant des similarités mineures. Cette mesure recoupe les deux précédentes pour $q = 1$ (Sum-Sum) et $q \to \infty$ (Max-Max).

Beaucoup de façons d'agréger les similarités mineures peuvent être imaginées. Il convient cependant de s'interroger sur les propriétés attendues de ces agrégations. Nous en listons ici quelques unes qui nous semblent essentielles ou d'autres souhaitables pour que la représentation en sacs-de-sacs garde une sémantique correcte. En RI, cette sémantique doit assurer l'ordonnancement des documents par proximité avec la requête. Les modèles usuels traduisent cette proximité selon une réponse graduelle, de manière à induire un ordre total entre les documents. L'agrégation des similarités mineures se doit donc de conserver au mieux cette propriété. Pour simplifier les notations, nous considérons l'agrégation comme une fonction, notée Aggreg, prenant en paramètre les similarité mineures notées a, b, c....

Associativité Cette propriété traduit le fait que le résultat d'une agrégation soit considérée de même nature qu'une similarité mineure et puisse être à son tour utilisée pour une agrégation. Cela nous permet de généraliser les propriétés ci-dessous, définies sur des relations binaires, à des fonctions n-aires : Aggreg(a, b, c) = Aggreg$(a,$ Aggreg$(b, c))$ = Aggreg(Aggreg$(a, b), c)$. Ainsi, on peut définir la fonction PowerScalar avec deux arguments : Aggreg$(a, b) = \sqrt[q]{a^q + b^q}$.

Commutativité. Comme nous l'avons dit, aucun ordre n'est à considérer pour les sacs-de-sacs ; on souhaite donc avoir une fonction commutative : Aggreg(a, b) = Aggreg(b, a)

Monotonie croissante. Une autre propriété essentielle est la monotonie de l'agrégation (croissante si la similarité mineure l'est) en fonction des similarités mineures :
$\forall a \geq b, c \geq d,$ Aggreg$(a, c) \geq$ Aggreg(b, d)

Certaines autres propriétés ne sont pas nécessaires pour que la métrique résultante ait bien le comportement attendu, mais peuvent être recherchées pour espérer de bons résultats.

Élément neutre. Pour ne pas favoriser les documents contenant beaucoup de sacs sans liens avec la requête, il est souhaitable que la mesure d'agrégation ait un élément neutre qui soit le minimum de la similarité mineure. C'est le cas avec les fonctions proposées en section 2.2 dont l'élément neutre est 0, avec une similarité mineure basée sur le produit scalaire : $Aggreg(a, 0) = a$

Continuité. La continuité de la fonction d'agrégation en fonction de toutes ses variables (similarités mineures) n'est pas non plus une condition nécessaire pour la sémantique de l'agrégation. Cependant, une telle continuité permet évidemment un comportement plus facilement interprétable et prévisible.

Fonctions archimédiennes. Il est souhaitable que l'ajout de similarités non nulles (supérieures à l'élément neutre) augmentent le score d'un document. Par analogie avec les groupes archimédiens, nous qualifions ces fonctions d'archimédiennes :
$\forall x, y \in]0, \infty[^2, \exists n \in \mathbb{N}+$ tel que $Aggreg(\underbrace{x, x, ..., x}_{n \text{ fois}}) > y$, avec 0 pour élément neutre.
Cette propriété est respectée par les fonctions Sum-Sum et PowerScalar (pour $q \neq \infty$), mais pas par la fonction Max-Max. Notons que cette propriété peut se restreindre à considérer les paires $x, y \in]0, l[^2$ où l est le maximum théorique de la similarité mineure, ce qui est utile pour une fonction qui serait également associative.

3 Expérimentation dans un cadre vectoriel

3.1 Contexte expérimental

Pour nos évaluations, nous utilisons une collection de RI appelée INIST, composée de 160 000 documents (résumés d'articles de diverses disciplines scientifiques) et de 30 requêtes et leurs jugements de pertinence (vérité terrain constituée manuellement). Les requêtes fournies sont composées de plusieurs champs : titre, corps, description et concepts associés. Nous évaluons classiquement nos résultats en terme de précision moyenne (MAP), et de précision (P@x) à différents seuils. Pour s'assurer que les différences constatées entre deux systèmes sont statistiquement significatives, nous utilisons un test de Wilcoxon ($p = 0.05$) (Hull, 1993).

Dans ces expériences et les suivantes rapportées dans cet article, les textes que nous traitons doivent être découpés en sous-documents, chacun étant représenté par un sac. Selon le formatage disponible (textes organisés en paragraphes ou non, etc.) et l'application visée, plusieurs options peuvent être explorées. Dans le cadre de nos évaluations, nous adoptons un découpage en phrase (détectées via les marques de ponctuations. Les requêtes sont également représentées en sacs de sacs : un sous-document correspond ici aussi à une phrase, s'il y en a, ou au plus à un champ (titre, corps...).

3.2 Adaptations des pondérations

Selon le modèle adopté, il peut être nécessaire d'adapter les pondérations utilisées. La plupart des fonctions de pondérations de RI font intervenir des valeurs calculées sur le document (IDF, longueur du document DL...). Le passage à une unité plus petite pose la question du calcul de ces valeurs. Nous rapportons ci-dessous une des expériences menées pour tester deux stratégies : dans un cas, les variables problématiques (IDF, DL...) sont calculées classiquement sur le document, et dans l'autre cas, sur le sous-document (le DL est alors la longueur du sous-document considéré, l'IDF est la fréquence document inverse dans l'ensemble des sous-documents de la collection). Pour ces expériences préliminaires, nous utilisons la collection INIST, nous fixons Sum-Sum comme mesure d'agrégation. La similarité mineure est le modèle Okapi-BM25 (Robertson et al., 1998), classiquement utilisé en RI, avec les constantes fixées à leur valeur par défaut : $k_1 = 2$, $k_3 = 1000$ et $b = 0.75$.

Les résultats sont donnés dans le tableau 1, avec la représentation sac-de-mots classique servant de base de comparaison. Ces résultats, confirmés sur d'autres collections, modèles et paramètres, soulignent la nécessité de prendre en compte le document dans son ensemble pour

	MAP	P@5	P@10	P@50	P@100
Okapi BoW	18.53	42.22	35.67	22.40	15.30
Okapi BoBoW sous-doc	-7.56 %	-10.83 %	-9.57 %	-3.44 %	*-0.44 %*
Okapi BoBoW doc	+3.88 %	+2.33 %	*+1.85 %*	*+0.76 %*	*-0.031 %*

TAB. 1: Performances des systèmes BoBoW par rapport au modèle BoW selon la granularité adoptée pour les cacul de l'IDF et du DL.

	MAP	P@5	P@10	P@50	P@100
BoW	18.53	42.22	35.67	22.40	15.30
BoBoW Sum-Sum	+3.88 %	+2.33 %	*+1.85 %*	*+0.76 %*	*-0.03 %*
BoBoW Max-Max	-4.50 %	*-0.82 %*	*+0.03 %*	*-0.93 %*	-5.4 %
BoBoW PowerScalar $q = 2$	+7.85 %	+2.54 %	+7.40 %	+5.22 %	+4.38 %
BoBoW PowerScalar $q = 3$	+6.49 %	+2.48 %	+5.88 %	+4.22 %	*+0.56 %*

TAB. 2: Performances des différents systèmes vectoriels.

calculer ces variables (IDF, DL...). Dans les expérimentations rapportées dans la suite de cet article, nous calculons donc ces valeurs à l'échelle du document complet.

3.3 Résultats

Nous évaluons les systèmes de RI BoBoW avec les paramètres décrits dans les sous-sections précédentes selon les fonctions d'agrégation présentées en section 2.2. Le tableau 2 présente les résultats obtenus respectivement en utilisant un modèle Okapi-BM25 (avec les paramètres IDF et DL calculés à l'échelle du document). Nous faisons apparaître les différences par rapport à la référence ; celles non statistiquement significatives sont en italiques.

On remarque que la représentation en sacs-de-sacs de mots obtient dans la plupart des cas des performances au moins équivalentes à celles du sac-de-mots classique, alors même que nous n'en avons optimisé aucun des paramètres. Cela montre le potentiel intéressant de ce type de représentations. À ce titre, les bons résultats de l'agrégation PowerScalar recoupent les constatations faites dans d'autres contextes (Ebadat et al., 2012; Gosselin et al., 2007). À l'inverse, le fait que Max-Max fonctionne moins bien que les autres agrégations, comme dans les tentatives de (Wilkinson, 1994), est dû au caractère non archimédien de cette fonction (la pertinence repose sur la proximité entre un seul sous-document du document et de la requête).

4 Conclusion

L'objectif de cet article était d'étudier les conditions nécessaires à l'utilisation des représentations en sac-de-sacs-de-mots, en s'intéressant notamment aux fonctions d'agrégation au cœur de cette approche. Nous avons mis en lumière quelques unes des propriétés souhaitables de ces fonctions, que nous avons illustrées dans un cadre vectoriel classique. Les expérimentations menées ont mis en exergue l'importance du choix de la fonction, en fonction notamment de son comportement seuillant ou non. Bien que notre but n'était pas d'optimiser un système

particulier, les très bons résultats de certaines configurations valident nos propositions et soulignent le potentiel de ces représentations.

Références

Bunke, H. (2000). Recent developments in graph matching. In *Proc. of 15th International Conference on Pattern Matching*, Barcelone, Espagne, pp. 117–2124.

Ebadat, A.-R., V. Claveau, et P. Sébillot (2012). Semantic Clustering using Bag-of-Bag-of-Features. In *Actes de la 9e conférence en recherche d'information et applications, CORIA 2012*, Bordeaux, France, pp. 229–244.

Gosselin, P., M. Cord, et S. Philipp-Foliguet (2007). Kernels on bags of fuzzy regions for fast object retrieval. In *IEEE International Conference on image processing, ICIP 2007*, Volume 1, pp. 177–180.

Haussler, D. (1999). Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, University of California at Santa-Cruz.

Hull, D. (1993). Using Statistical Testing in the Evaluation of Retrieval Experiments. In *Proc. of the 16th Annual ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'93*, Pittsburgh, États-Unis.

Kondor, R. et T. Jebara (2003). A kernel between sets of vectors. In A. Press (Ed.), *Proc. of the 12th International Conference on Machine Learning (ICML)*, Washington DC, USA.

Luk, R. W., H. Leong, T. S. Dillon, A. T. Chan, W. B. Croft, et J. Allan (2002). A survey in indexing and searching xml documents. *Journal of the American Society for Information Science and Technology 53*(6), 415–437.

Robertson, S. E., S. Walker, et M. Hancock-Beaulieu (1998). Okapi at TREC-7 : Automatic Ad Hoc, Filtering, VLC and Interactive. In *Proc. of the 7th Text Retrieval Conference, TREC-7*, pp. 199–210.

Wilkinson, R. (1994). Effective retrieval of structured documents. In W. B. Croft et C. J. van Rijsbergen (Eds.), *Proc. of the 17th Annual International ACM-SIGIR Conference*, Dublin, Ireland, pp. 311–317. ACM/Springer.

Summary

In this paper, the interest of representing texts as bags-of-bags-of-words is studied. At the heart of it, the similarity computation between two documents requires to evaluate the similarity between each possible pair of bags. The performance of this representation is evaluated in a standard IR framework, and the properties of the aggregation functions are discussed. The experiments reported show the interest of this representation when the aggegation has specific properties, which then yields important performance gains.