GENDESC : Vers une nouvelle représentation des données textuelles

Guillaume Tisserant *, Violaine Prince *, Mathieu Roche *,**

* LIRMM, CNRS, Université Montpellier 2
161 rue Ada, 34095 Montpellier Cedex 5, France
tisserant@lirmm.fr, prince@lirmm.fr, mroche@lirmm.fr
** TETIS, Cirad, Irstea, AgroParisTech
500 rue Jean-François Breton, 34093 Montpellier Cedex 5, France
mathieu.roche@cirad.fr

Résumé. Dans cet article, nous nous intéressons à la classification automatique de données textuelles par des algorithmes d'apprentissage supervisé. L'objectif est de montrer comment l'amélioration de la représentation des données textuelles influe sur les performances des algorithmes d'apprentissage. Partant du postulat qu'un mot n'a pas un sens bien établi sans son contexte, nous proposerons des descripteurs donnant le plus d'information possible sur le contexte des mots. Pour cela, nous avons mis au point une méthode, nommée GENDESC, qui consiste à "généraliser" les mots les moins pertinents pour la classification, c'est-à-dire, à éviter le bruit sémantique (souvent dû à la polysémie) provoqué par ces termes non ou peu pertinents. Cette généralisation s'appuie sur des informations grammaticales, telles que la catégorie et la position dans la structure. La méthode GENDESC a été évaluée et adaptée à la problématique de classification de textes selon une opinion ou une thématique.

1 Introduction

La problématique à laquelle cet article se confronte, est liée à la tâche de classification de données textuelles. Les données textuelles sont extrêmement difficiles à analyser et classifier d'après Witten et Frank (2005). Les algorithmes d'apprentissage supervisé que nous nous proposons d'utiliser dans cette étude nécessitent de connaître la classe (e.g. thème, sentiment, etc.) à associer à chaque document. Les entrées de ces algorithmes sont des "paquets" de descripteurs linguistiques (c'est-à-dire des critères de classification issus des propriétés du matériau langagier, comme la catégorie grammaticale, la fonction lexicale, le rôle syntaxique, etc. mais aussi des critères terminologiques) représentant le document à classifier. Une fois la phase d'apprentissage effectuée, le modèle appris peut attribuer une classe à des "paquets de descripteurs" non étiquetés qui lui sont donnés. Un résumé ce cette approche est donné en Figure 1. La qualité de la classification proposée par l'algorithme va donc dépendre à la fois de la qualité de l'algorithme d'apprentissage, mais aussi de la façon dont les données qui lui sont transmises sont représentées, comme le montre Béchet (2009).

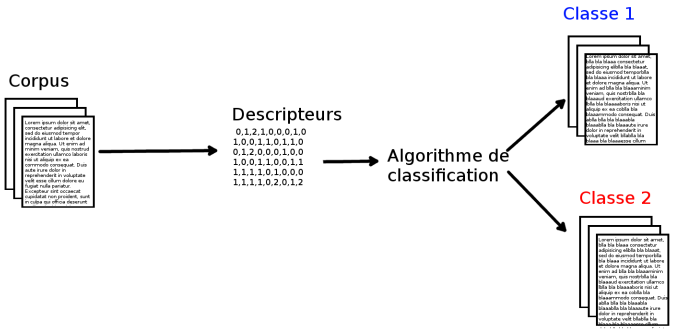

FIG. 1 – *Processus global.*

Dans l'abondante littérature autour de la fouille de textes, la méthode classique de représentation de données textuelles est le modèle "sac de mots" : les mots eux-mêmes sont considérés comme des descripteurs (les critères sont alors uniquement terminologiques), utilisés en tant qu'entrées des algorithmes d'apprentissage (Salton et McGill (1986)). En dépit de sa popularité, cette méthode possède de nombreuses limites. Tout d'abord, elle fait ressortir un très grand nombre de descripteurs, autant qu'il y a de *termes* (un terme étant vu comme une entrée d'un dictionnaire, et donc toutes les formes lexicales relevant d'une même entrée, sont comptabilisées comme appartenant au même terme) dans un document. Ensuite, elle perd toutes les informations liées à la place des mots et à leurs rôles syntaxiques dans la phrase, ainsi que toutes les informations liées au contexte. En effet les rôles syntaxiques permettent une mise en perspective de l'importance des mots : certains sont *gouverneurs*, et donc dominants dans la phrase, et d'autres sont *gouverné*s, et plus en arrière-plan. Les travaux présentés dans cet article soulignent donc que les données textuelles, qui sont par nature complexes, doivent être représentées par des structures plus subtiles dans un processus de fouille de données. Assez vite, il a été mis en avant l'utilité d'avoir des descripteurs plus généraux que les mots, en particulier pour ceux dont la catégorie est par exemple plus importante que le terme lui-même. Des expérimentations ont montré la possibilité d'utiliser le radical ou le lemme de chaque mot (Porter (1980)), ou le *POS-Tag* (catégorie grammaticale) (Gamon (2004)). Pour des besoins de classification sur des tâches spécifiques, certains travaux (Joshi et Penstein-Rosé (2009)) proposent de généraliser certains mots spécifiques (déterminés en fonction de leur rôle grammatical) en les remplaçant par un descripteur plus général. Nous proposons ici une méthode de généralisation indépendante de la tâche, permettant d'améliorer la qualité de la classification de données textuelles sans connaissance *a priori* sur la tâche de classification.

Nous rappelons, dans un premier temps, les différents travaux de recherche sur les méthodes de description des données textuelles adaptées à l'apprentissage supervisé (Section 2). En Section 3 et 4, nous proposons de nouvelles méthodes de construction et sélection de descripteurs qui sont évaluées en Section 5. Enfin, nous dressons le bilan actuel de nos travaux et présentons quelques perspectives en Section 6.

2 État de l'art

Les données textuelles sont particulièrement complexes à interpréter et à classifier de manière automatique (Witten et Frank (2005)). Outre les différents algorithmes de classification de textes (Aggarwal et Zhai (2012)), de nombreux travaux étudient l'influence des représentations des données textuelles sur ces derniers (Béchet (2009)). Les travaux que nous proposons ici se concentrent sur une telle problématique.

Dans la suite de cette section, nous allons considérer une phrase exemple (cf. Figure 2) et indiquer, pour chaque type de traitement, la façon dont la phrase exemple serait représentée.

```
Meilleur film et meilleur acteur pour Cria Cuervos :
Cannes est un des meilleurs festivals.
```

FIG. 2 – *Phrase exemple.*

Le modèle le plus classique de représentation d'un document textuel est le modèle sac de mots (Salton et McGill (1986)). Dans ce modèle, chaque mot est considéré comme un descripteur indépendant. Pour la phrase exemple, une représentation sac de mots donnerait :

Meilleur	film	et	meilleur	acteur	pour	Cria
Cuervos	Cannes	est	un	des	meilleurs	festivals

FIG. 3 – *Sac de mots issu de la phrase exemple.*

Cette représentation fait perdre toutes les informations liant les mots à leur contexte, elle a pourtant l'avantage d'être simple et très généraliste.

La perte de l'information concernant la place du mot dans la phrase peut être compensée par l'introduction de n-grammes de mots. Nous pouvons définir un n-gramme comme une séquence de n éléments consécutifs (Pratt (1996)). Bien que les premiers éléments utilisés étaient les lettres, nous pouvons parfaitement prendre en compte des n-grammes de mots comme dans de nombreux travaux en fouille de textes (Fürnkranz (1998); Béchet (2009)). On appelle les n-grammes composés de deux descripteurs des bigrammes, et ceux composés de trois descripteurs des trigrammes. Il a été montré par Fürnkranz (1998) que les bigrammes et trigrammes de mots étaient plus efficaces que des n-grammes de taille supérieure. La phrase exemple représentée sous forme de bigrammes de mots donnerait le résultat présenté en Figure 4.

Une telle représetation met en relief une partie du contexte du mot. Par exemple le bigramme `Meilleur-film` permet de savoir qu'il existe les mots `Meilleur` et `film` qui sont voisins. Toutefois, cette solution génère des descripteurs moins généraux que les mots. Le descripteur `Meilleur-film` est par exemple un descripteur clairement plus spécifique que les mots `Meilleur` ou `film`. Cela peut poser problème, entre autres pour les tâches de classification car

Meilleur-film	film-et	et-meilleur	meilleur-acteur
acteur-pour	pour-Cria	Cria-Cuervos	Cuervos-Cannes
Cannes-est	est-un	un-des	des-meilleurs
meilleurs-festivals			

FIG. 4 – *Bigrammes issus de la phrase exemple.*

les algorithmes de classification doivent s'appuyer sur des descripteurs suffisamment représentatifs. Il est donc important que les descripteurs apparaissent dans un nombre de documents significatif pour que les algorithmes puissent apprendre efficacement.

Les n-grammes ont donc souvent été utilisés avec des données généralisées, comme Gamon (2004) qui propose de construire des n-grammes de POS-tag. Un POS-tag (Part-Of-Speech tagging), traduit en français étiquetage morpho-syntaxique, est le processus qui consiste à associer aux mots d'un texte ses informations grammaticales comme dans l'exemple de la Figure 5.

```
Meilleur/ADJ film/NOM et/KON meilleur/ADJ acteur/NOM
pour/PRP Cria/NAM Cuervos/NAM :/PUN Cannes/NAM
est/VER un/DET des/PRP meilleurs/ADJ festivals/NOM
```

FIG. 5 – *Phrase exemple étiquetée.*

Cet étiquetage peut se révéler tout à fait crucial pour la classification. Par exemple, la présence d'un certain nombre d'adverbes ou d'adjectifs est une information prépondérante pour la classification de données d'opinion (Xia et Zong (2010)). L'étiquetage grammatical (cf. Figure 5) permet également de construire des n-grammes de POS-tag comme dans l'exemple de la Figure 6. Cette généralisation permet d'avoir des informations synthétiques tout à fait pertinentes pour identifier des structures syntaxiques récurrentes utiles pour certaines tâches comme par exemple la correction orthographique (Grouin (2008)). Cependant, pour la classification thématique qui représente le cœur de nos travaux, ce type de structure est clairement inadaptée.

```
/ADJ-/NOM /NOM-/KON /KON-/ADJ /ADJ-/NOM /NOM-/PRP
/PRP-/NAM /NAM-/NAM /NAM-/PUN /PUN-/NAM /NAM-/VER
/VER-/DET /DET-/PRP /PRP-/ADJ /ADJ-/NOM
```

FIG. 6 – *Bigrammes de POS-tag.*

Porter (1997) propose quant à lui une solution alternative mais tout à fait complémentaire en utilisant la forme canonique des mots comme descripteur. La forme canonique des mots est la forme dans laquelle un mot est trouvé dans les dictionnaires. Cette forme s'oppose à la forme fléchie où le mot peut apparaître sous une autre forme (les verbes peuvent être conjugués, les substantifs accordés, etc.). L'utilisation de la forme canonique permet d'avoir des descripteurs

un peu plus généraux que la forme fléchie du mot, mais en conservant une partie importante de l'information sémantique comme dans l'exemple ci-dessous :

```
Meilleur film et meilleur acteur pour Cria Cuervos :
Cannes être un du meilleur festival.
```

FIG. 7 – *Phrase exemple avec les mots sous leur forme canonique.*

Notons que de telles généralisations peuvent aussi être prises en compte dans des n-grammes.

Le principal problème de la généralisation concerne l'importante perte d'information. La généralisation sous forme de POS-tag fait par exemple perdre toute information sémantique. Pour permettre un apprentissage optimal, il semble donc nécessaire de conserver certains mots et d'en généraliser d'autres.

Plusieurs travaux ont vu le jour dans ce sens. Par exemple, Joshi et Penstein-Rosé (2009) proposent de ne considérer que les bigrammes dans lesquels les deux mots sont en relation et de généraliser le mot "gouverneur" de la relation pour effectuer de la classification de sentiments. Une telle approche donnerait le résultat ci-dessous à partir de notre phrase exemple :

Bigrammes liés aux sentiments : `Meilleur/ADJ-film/NOM meilleur/ADJ-acteur/NOM meilleurs/ADJ-festivals/NOM`
Bigrammes liés aux sentiments généralisés : `Meilleur-/NOM meilleur-/NOM meilleurs-/NOM`

FIG. 8 – *Les n-grammes de mots de relations et la version généralisée (Joshi et Penstein-Rosé (2009))*

D'autres approches associent les informations morpho-syntaxiques à des connaissances sémantiques. Par exemple, Nasukawa et Yi (2003) utilisent des dictionnaires de sentiments constitués manuellement et une détection de relations syntaxiques à partir des textes pour effectuer de la classification de sentiments. Ce type de méthode permet de classer efficacement les textes contenant un ou plusieurs mots du dictionnaire de sentiment mais se retrouve limité face à des phrases n'en contenant pas.

Notons aussi qu'il existe des approches de classification de textes construites autour de méthodes statistiques. Par exemple, LSA (Landauer et Dumais (1997); Roche et Kodratoff (2003)) donne davantage de poids à certains mots du contexte en effectuant une "approximation numérique" par décomposition en valeurs singulières. Ce type d'approche peut être combiné avec des méthodes syntaxiques et/ou sémantiques pour améliorer les performances (Béchet et al. (2008)).

Synthèse de l'état de l'art

Pour résumer, dans les travaux antérieurs, nous pouvons mettre en avant deux types d'orientations pour représenter les données textuelles pour les tâches de classification de textes. Dans

	forme fléchie	lemmes	POS-tag	n-grammes
Shannon (1948)				×
Salton et McGill (1986)	×			
Porter (1980)		×		
Gamon (2004)			×	×
Joshi et Penstein-Rosé (2009)	×		×	×

TAB. 1 – *Types de descripteur utilisés*

	Méthode morphosyntaxique	Méthode statistique	Méthode sémantique
Nasukawa et Yi (2003)	×		×
De Melo et Siersdorfer (2007)			×
Plantié et al. (2008)	×	×	
Roche et Kodratoff (2003)		×	
Landauer et Dumais (1997)		×	
Béchet et al. (2008)	×	×	×
Joshi et Penstein-Rosé (2009)	×		
Xia et Zong (2010)	×		

TAB. 2 – *Méthodes de généralisation*

un premier temps, nous pouvons noter plusieurs méthodes de généralisation en considérant les mots originaux des données textuelles (mots sous forme fléchie) jusqu'à une forme très générale (fonctions grammaticales). Par ailleurs, l'utilisation de groupes de mots se révèle pertinent pour certaines tâches de classification (Joshi et Penstein-Rosé (2009); Gamon (2004)), en particulier lorsqu'ils sont associés à d'autres types de descripteurs linguistiques de base (Plantié et al. (2008)). Ces différents types d'approches sont résumés dans la Table 1. Les travaux de la littérature et la synthèse de la Table 1 montrent que de nombreuses méthodes combinent efficacement différents descripteurs pour des tâches de classification de textes. Toutefois, lorsqu'il y a un choix de descripteur à établir, la majorité des méthodes sont construites en fonction de la tâche à réaliser. Notre approche se différencie des autres par le fait qu'elle peut s'appliquer à tous types de classification, sans connaissances *a priori* du corpus à traiter.

Par ailleurs outre les caractéristiques des descripteurs utilisés en classification, nous pouvons identifier plusieurs types de méthodes de généralisation que nous pouvons regrouper en trois catégories : méthodes morphosyntaxiques, statistiques et sémantiques. La Table 2 qui synthétise les différentes approches permet de mettre en avant l'importance des informations morpho-syntaxiques pour la classification de textes. Bien que différentes approches exploitent des informations sémantiques pour la classification de documents (Nasukawa et Yi (2003); De Melo et Siersdorfer (2007)), notre proposition, présentée en Section 3, n'exploite pas ce type d'information afin de rester le plus générique possible.

3 Vers une nouvelle représentation des données textuelles

3.1 Principe et motivations

L'idée de remplacer certains mots par leur catégorie grammaticale vient du constat suivant : si pour une tâche donnée, certains mots caractéristiques d'une classe doivent être utilisés tels quels en tant que descripteurs, certains autres n'apportent que très peu d'information. Ces derniers représentant, par exemple, des mots rares et/ou spécifiques qui seront généralisés par notre approche GENDESC (Tisserant et al. (2013)). Nous proposons de généraliser certains mots à travers leur étiquette grammaticale, celle-ci pouvant se révéler particulièrement intéressante (Xia et Zong (2010)).

L'approche que nous proposons se décline en différentes étapes :

1. La première détermine la fonction grammaticale de chaque mot du corpus.
2. La suivante consiste à sélectionner les mots à généraliser. Cette sélection s'appuie sur différentes mesures statistiques données dans la Section suivante.
3. Les descripteurs sélectionnés seront utilisés directement sous leur forme fléchie.
4. Nous proposons ensuite de construire des unigrammes, bigrammes et trigrammes à partir des mots conservés et des étiquettes des mots généralisés. Ces données vont constituer nos descripteurs utilisés pour entraîner les algorithmes d'apprentissage.

La Section suivante détaille l'intégralité du processus mis en œuvre.

3.2 GENDESC : Généralisation partielle des descripteurs

L'approche GENDESC consiste à généraliser les mots dont la catégorie grammaticale est une information plus intéressante que le mot en lui-même pour l'étape de classification. Pour choisir les mots que nous allons généraliser, nous avons choisi de commencer par définir une fonction de rang associant un poids à chaque mot, indiquant sa pertinence qui sera évaluée pour une tâche de classification. Les mots dont le poids est inférieur à un seuil donné sont ensuite généralisés par leur étiquette grammaticale. Nous allons présenter différentes fonctions de rang et donner des exemples fondés sur la problématique de la classification d'opinion. Le but d'une telle classification est d'associer chaque phrase à l'opinion qu'elle véhicule. Le *Corpus exemple* donné en Figure 9 est constitué de 9 phrases étiquetées comme *positives* ou *négatives*.

3.2.1 La mesure TF

La fonction de rang de base s'appuie sur la notion de fréquence, elle est notée TF (term frequency) et correspond à la fréquence d'apparition d'un terme dans un document. Elle est construite autour de l'idée que plus un mot est représenté dans un document, plus il est représentatif de ce document.

$$TF(x, y) = \frac{\text{Nombre d'apparitions du mot } x \text{ dans le document } y}{\text{Nombre de mots dans le document } y} \quad (1)$$

1. `Fight club est un mauvais film.` *Négatif*
2. `Ce CD est meilleur que tous les autres.` *Positif*
3. `Le meilleur court métrage qu'on ait pu voir dans ce festival.` *Positif*
4. `Le plus mauvais moment de la série, tellement ennuyeux...` *Négatif*
5. `Cette musique est vraiment bonne.` *Positif*
6. `Ce n'est pas un bon film, il est vraiment trop ennuyeux.` *Négatif*
7. `Meilleur film et meilleur acteur pour "Cria Cuervos" : Cannes est un des meilleurs festivals!` *Positif*
8. `Le nouveau Green lantern est un mauvais Comics : écrit par un mauvais scénariste et dessiné par un mauvais illustrateur.` *Négatif*
9. `Le film du moment, le film de l'année, le film du siècle!` *Positif*

FIG. 9 – *Corpus exemple*

Par exemple, si nous prenons les mesures des mots `Cannes` et `meilleur` dans la phrase 7 du Corpus exemple, on obtient :

$$TF(\texttt{Cannes}, 7) = \frac{1}{14} \approx 0.071$$

$$TF(\texttt{meilleur}, 7) = \frac{2}{14} \approx 0.143$$

Nous pouvons relever que le mot `meilleur` qui se révèle particulièrement important dans la phrase, a une mesure assez élevée. Toutefois, la fonction TF va aussi retourner des valeurs élevées aux mots outils (*stop words*), ou à certains mots très présents dans le corpus qui ne sont pas des marqueurs de classe comme les mots `le` et `film` dans la phrase 9 du Corpus exemple (cf. Figure 9) :

$$TF(\texttt{le}, 9) = \frac{3}{14} \approx 0.214$$

$$TF(\texttt{film}, 9) = \frac{3}{14} \approx 0.214$$

Pour cette raison, TF, dans la plupart des cas, est pondéré par une autre mesure. Notons par ailleurs que, dans des documents courts comme dans le Corpus exemple où chaque document est une phrase, la plupart des mots importants n'apparaissent qu'une seule fois dans le document. Il est donc nécessaire d'utiliser des mesures plus appropriées à une telle situation.

3.2.2 Les mesures DF et IDF

Pour pallier les limites de la mesure précédente, une autre fonction très couramment utilisée est DF (Document Frequency - cf. Formule 2). Cette dernière correspond au nombre

d'apparitions d'un mot dans l'ensemble du corpus. Elle se fonde sur l'idée que plus un mot apparaît dans le corpus, plus il est intéressant pour décrire des documents appartenant à ce corpus.

$$DF(x) = \text{Nombre de documents où le mot } x \text{ apparaît} \tag{2}$$

A l'inverse, la mesure IDF (Inverse Document Frequency - cf. Formule 3) part de l'idée que les mots les plus informatifs d'un document au sein d'un corpus ne sont pas ceux apparaissant le plus dans ce corpus. Ces mots sont la plupart du temps des mots outils ou des termes propres au corpus que l'on trouvera dans beaucoup de documents du corpus sans qu'ils nous donnent d'information sur celui-ci.

$$IDF(x) = \log \frac{\text{Nombre de documents}}{DF(x)} \tag{3}$$

Si nous reprenons les trois mots exemples donnés dans la sous-section précédente, nous obtenons les valeurs ci-dessous :

$$DF(\texttt{meilleur}) = 3$$

$$IDF(\texttt{meilleur}) = \log \frac{9}{3} \approx 0.477$$

$$DF(\texttt{le}) = 4$$

$$IDF(\texttt{le}) = \log \frac{9}{4} \approx 0.811$$

$$DF(\texttt{film}) = 4$$

$$IDF(\texttt{film}) = \log \frac{9}{4} \approx 0.811$$

$$DF(\texttt{Cannes}) = 1$$

$$IDF(\texttt{Cannes}) = \log \frac{9}{1} \approx 2.197$$

Nous pouvons constater que `meilleur`, mot intéressant pour la classification d'opinion, a une valeur DF plus élevée que des mots quelconques du corpus, comme `Cannes`. Aussi, comme TF, cette mesure a tendance à donner des scores importants aux mots outils et au vocabulaire général du domaine du corpus (par exemple, `film`). Ce type de mot est peu significatif pour des tâches liées à la classification d'opinion.

À l'inverse, IDF donne un score très faible aux mots outils et aux mots très présents dans l'ensemble des classes du corpus. De plus, une telle mesure de *discriminance* donne également de faibles scores aux mots comme `meilleur` qui apparaissent de nombreuses fois dans la même classe. Il peuvent pourtant se révéler très utiles pour la classification d'opinion.

3.2.3 La mesure $TF - IDF$

La fonction la plus classique combine les deux critères présentés précédemment, à savoir les mesures TF et IDF, pour constituer la fonction $TF - IDF$ (term frequency - inverse document frequency - cf. Formule 4). Le principe de cette approche est de pondérer la méthode fréquentielle TF par le nombre de documents dans lesquels un terme cible apparaît. Elle part de l'idée qu'un mot apparaissant beaucoup dans un document mais peu dans le reste du corpus est représentatif de ce document.

$$TF - IDF(x, y) = TF(x, y) \cdot IDF(x) \tag{4}$$

Si nous calculons les mesures des mots précédents avec une telle pondération, nous obtenons :

$$TF - IDF(\texttt{Cannes}, 7) = \frac{1}{14} \cdot \log \frac{9}{1} \approx 0.157$$

$$TF - IDF(\texttt{meilleur}, 7) = \frac{2}{14} \cdot \log \frac{9}{3} \approx 0.157$$

$$TF - IDF(\texttt{un}, 7) = \frac{1}{14} \cdot \log \frac{9}{4} \approx 0.058$$

$$TF - IDF(\texttt{film}, 7) = \frac{1}{14} \cdot \log \frac{9}{4} \approx 0.058$$

Comme nous pouvons le constater, les mots rares dans le corpus ou présents plusieurs fois dans un même document ont des scores assez supérieurs aux mots outils ou aux mots généraux du corpus.

3.2.4 La mesure D

La fonction de Discriminance, que nous appelons D (cf. Formule 5), est construite sur les mêmes principes généraux que ceux véhiculés par la mesure $TF - IDF$. La différence tient au fait que la fréquence d'apparition dans un document (TF) est remplacée par la fréquence d'apparition dans la classe qui contient le plus souvent le mot cible. En effet, le nombre d'apparitions du mot dans une classe, pondéré par son nombre d'apparitions dans l'ensemble du corpus, semble une mesure adaptée à la problématique de classification de documents. Ceci s'explique par le fait qu'un tel principe met en relief l'importance du mot au regard des classes constituées.

$$D(x) = \frac{\text{nombre d'occurrences du mot } x \text{ dans la classe qui le contient le plus}}{\text{Nombre d'occurrences du mot dans l'intégralité du Corpus}} \tag{5}$$

Si nous calculons la mesure D de nos trois mots exemples, nous obtenons :

$$D(\texttt{Cannes}) = \frac{1}{1} = 1$$

$$D(\texttt{meilleur}) = \frac{4}{4} = 1$$

$$D(\texttt{le}) = \frac{4}{6} \sim 0.667$$

$$D(\texttt{film}) = \frac{4}{6} \sim 0.667$$

Les mots permettant d'associer un document à une classe, comme `meilleur` obtiennent des scores élevés. Cette mesure donne aussi des scores importants à certains mots rares qui pourraient se trouver répartis de manière inéquitable dans les données d'apprentissage sans que ce soit le cas dans les données de test. Pour éviter cet écueil, il est possible de combiner cette mesure à DF, mais cela n'a pas amélioré les résultats dans les expérimentations que nous avons menées.

3.3 Un exemple de généralisation partielle

Dans cette section, nous proposons de détailler un exemple complet consistant à généraliser certains mots sélectionnés selon les mesures précédemment décrites. Prenons en exemple le corpus de la Figure 9. Dans ce contexte, le but de la classification est de prédire la polarité à associer à chaque document (phrase). À partir de l'exemple de la Figure 9, nous obtenons pour chaque mot le nombre de documents le contenant (cf Table 3).

Mot	**Nombre de documents contenant le mot**
`est`	6
`un`	5
`mauvais`	3
`film`	3
`ce`	3
`le`	3
`meilleur`	2
`festival`	2
`vraiment`	2
`et`	2
`bon`	2

TAB. 3 – *Nombre de documents contenant le mot pour les mots apparaissant plus d'une fois dans le corpus*

Ensuite, pour chaque mot de la phrase, la fonction grammaticale nous est donnée comme dans l'exemple ci-dessous :

```
Meilleur/ADJ film/NOM et/KON meilleur/ADJ acteur/NOM pour/PRP Cria/NOM
Cuervos/NOM :/PON Cannes/NOM est/VER un/DET des/PRP meilleurs/ADJ
festivals/NOM
```

Mot	Étiquette attachée au mot	Nombre de documents contenant le mot	Descripteur GENDESC
Meilleur	ADJ	3	Meilleur
film	NOM	3	film
et	KON	2	KON
meilleur	ADJ	3	meilleur
acteur	NOM	3	acteur
pour	PRP	1	PRP
Cria	NOM	1	NOM
Cuervos	NOM	1	NOM
:	PON	1	PON
Cannes	NOM	1	NOM
est	VER	6	est
un	DET	4	un
des	DET	1	DET
meilleurs	ADJ	1	ADJ
festival	NOM	2	NOM
!	PON	1	PON

TAB. 4 – *Application de la généralisation* GENDESC

Dans cet exemple, les étiquettes correspondent à :
ADJ : Adjectif ;
NOM : Nom ;
KON : Conjonction ;
PRP : préposition ;
VER : Verbe ;
DET : Déterminant ;
ADJ : Adjectif ;
PON : Ponctuation.

Nous pouvons ensuite utiliser les fonctions de rang définies en Section précédente. La fonction utilisée dans le premier exemple est le nombre de documents du corpus contenant le mot (c'est-à-dire la mesure DF). Pour cette mesure, le seuil représente donc le nombre minimum de documents présents dans le corpus qui doivent contenir le mot pour que celui-ci ne soit pas généralisé.

Pour la phrase exemple, si un seuil est fixé à 3, tous les mots apparaissant dans moins de 3 documents seront remplacés par leur étiquette grammaticale (cf. Tableau 4). La phrase initiale de notre exemple est alors généralisée de la manière suivante pour un seuil fixé à 3 :
`Meilleur film KON meilleur acteur PRP NOM NOM PON NOM est un DET ADJ NOM PON`

Dans ce contexte, plus le seuil est élevé, plus le nombre de mots généralisés est important. Nous détaillons ci-dessous le même exemple avec tous les seuils possibles de généralisation :

— Seuil à 1 (tous les mots sont conservés) : `Meilleur film et meilleur acteur pour "Cria Cuervos" : Cannes est un bon festival!`
— Seuil à 2 : `Meilleur film et meilleur acteur PRP NOM NOM PON NOM est un DET ADJ festival PON`
— Seuil à 3 : `Meilleur film KON meilleur acteur PRP NOM NOM PON NOM est un DET ADJ NOM PON`
— Seuil à 4 : `ADJ NOM KON ADJ NOM PRP NOM NOM PON NOM est un DET ADJ NOM PON`
— Seuil à 5 : `ADJ NOM KON ADJ NOM PRP NOM NOM PON NOM est DET DET ADJ NOM PON`
— Seuil à 7 (tout les mots sont remplacés) : `ADJ NOM KON ADJ NOM PRP NOM NOM PON NOM VER DET DET ADJ NOM PON`

Nous constatons que jusqu'à un certain point de généralisation, l'information pertinente (sur le type de relation que donne la phrase) devient plus évidente que dans la phrase entière. Les mots les moins pertinents sont généralisés en premier, alors que les mots indiquant l'appartenance à une classe comme `meilleur` ne sont généralisés qu'assez tard. Notons qu'un post-filtrage permet de supprimer les mots outils. Nous montrerons en Section 5 que certaines mesures comme D se révèlent en fait plus efficaces.

4 GENDESC et n-grammes

Dans cette section, nous allons nous intéresser à l'extension de GENDESC aux n-grammes.

4.1 n-grammes de mots

Un n-gramme de mots correspond à la concaténation de n mots voisins. Les signes de ponctuation peuvent être des informations contextuelles très importantes. Nous considérons donc ces marqueurs linguistiques comme de possibles éléments constituant les n-grammes au même titre que les mots. Prenons l'exemple de la Figure 2. En utilisant les mots comme descripteurs initiaux pour construire les n-grammes à partir de cet exemple, une représentation sous forme de 2-grammes (ou bigrammes) fournit le résultat ci-dessous :

```
Meilleur-film film-et et-meilleur meilleur-acteur acteur-pour
pour-Cria Cria-Cuervos Cuervos-: :-Cannes Cannes-est est-un
un-bon bon-festival festival-!
```

La représentation sous forme de 3-grammes (ou trigrammes) est donnée ci-dessous :

```
Meilleur-film-et film-et-meilleur et-meilleur-acteur
meilleur-acteur-pour acteur-pour-Cria pour-Cria-Cuervos
Cria-Cuervos-: Cuervos-:-Cannes :-Cannes-est Cannes-est-un
est-un-des un-des-meilleurs des-meilleurs-festival meilleurs-festival-!
```

Les n-grammes de mots sont très riches en information, car chaque descripteur contient à la fois un mot et une partie de son contexte. Mais ce sont des descripteurs textuels parfois trop spécifiques ayant peu de chance d'être présents dans de nombreux documents.

4.2 n-grammes d'étiquettes grammaticales

Outre les n-grammes de mots, il peut être utile d'identifier des n-grammes d'étiquettes grammaticales. À partir du même jeu de données, nous obtenons alors les résultats ci-dessous :

— Bigrammes d'étiquettes :
`ADJ-NOM NOM-KON KON-ADJ ADJ-NOM NOM-PRP PRP-NOM NOM-NOM NOM-PON PON-NOM NOM-VER VER-DET DET-DET DET-ADJ ADJ-NOM NOM-PON`

— Trigrammes d'étiquettes :
`ADJ-NOM-KON NOM-KON-ADJ KON-ADJ-NOM ADJ-NOM-PRP NOM-PRP-NOM PRP-NOM-NOM NOM-NOM-PON NOM-PON-NOM PON-NOM-VER NOM-VER-DET VER-DET-DET DET-DET-ADJ DET-ADJ-NOM ADJ-NOM-PON`

Les n-grammes d'étiquettes grammaticales sont plus généraux et ont donc de meilleures chances de se retrouver dans de nombreux documents. Mais l'information donnée est moins pertinente d'un point de vue sémantique pour décrire le document (cf. section 2).

4.3 n-grammes et généralisation partielle

Cette section décrit l'application de GENDESC aux n-grammes. Ceci revient à généraliser certains mots des n-grammes constitués en appliquant le même processus décrit précédemment. Un tel processus consiste, dans un premier temps, à appliquer la généralisation partielle avec GENDESC, puis à rechercher les n-grammes à partir de la représentation textuelle obtenue. Ce processus est illustré en Figure 10.

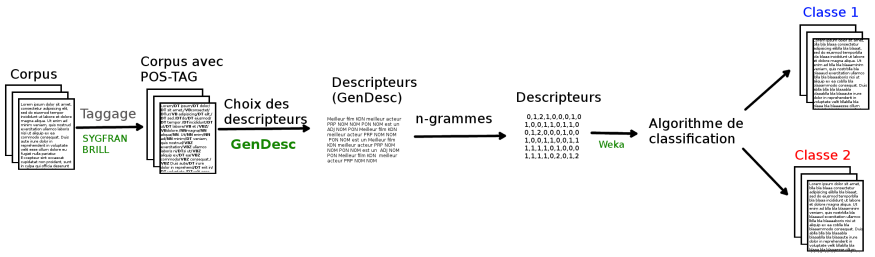

FIG. 10 – GENDESC *et n-grammes*

Par exemple, considérons la généralisation partielle ci-dessous :
`Meilleur film KON meilleur acteur PRP NOM NOM PON NOM est un ADJ NOM PON`

Nous obtenons les bigrammes ci-dessous :
`Meilleur-film film-KON KON-meilleur meilleur-acteur acteur-PRP PRP-NOM NOM-NOM NOM-PON PON-NOM NOM-est est-un un-DET DET-ADJ ADJ-NOM NOM-PON`

Les trigrammes sont donnés ci-dessous :
`Meilleur-film-KON film-KON-meilleur KON-meilleur-acteur meilleur-acteur-PRP acteur-PRP-NOM PRP-NOM-NOM NOM-NOM-PON NOM-PON-NOM PON-NOM-est NOM-est-un est-un-DET un-DET-ADJ DET-ADJ-NOM ADJ-NOM-PON`

Nous obtenons ainsi des n-grammes pouvant être composés soit de mots, soit d'étiquettes grammaticales, soit d'une combinaison des deux. Ayant généralisé les mots que nous pensons moins pertinents pour la classification, nous évaluons dans la section suivante l'ensemble de nos propositions.

5 Expérimentations

Dans cette section, nous présentons les expérimentations qui permettent de discuter l'utilisation des différentes fonctions de rang selon les types de représentations (unigrammes, bigrammes et trigrammes). Enfin, les résultats obtenus avec différents algorithmes d'apprentissage seront présentés.

5.1 Problématique et protocole expérimental

Dans ces travaux, nous avons choisi de nous intéresser à deux problématiques de classification afin de prédire (1) une opinion, (2) un parti politique.

— Pour la première problématique, le corpus utilisé est DEFT 2007 (Grouin et al. (2007)). Il est composé d'interventions de députés portant sur le vote de lois en examen à l'Assemblée Nationale. L'objectif est de classifier les documents selon que l'élu soit pour ou contre le projet de loi discuté. Nous avons travaillé à partir d'un sous-ensemble représentatif du corpus, composé de 1000 textes, équitablement répartis entre ceux contre les projets de loi, et les textes les défendant.
— Nous avons aussi expérimenté notre méthode sur un corpus de 1500 tweets équitablement répartis entre 5 partis politiques. L'objectif est de classer les tweets en fonction du parti du locuteur. Ce corpus sera appelé TWEET.

Les deux corpus ont été préalablement étiquetés grammaticalement grâce à l'analyseur morpho-syntaxique SYGFRAN (Chauché (1984)).

Nous avons testé 3 algorithmes d'apprentissage différents : un algorithme de classification bayésien, les arbres de décision, et un algorithme fondé sur les SVM (Support Vector Machine). Les 3 algorithmes ont été testés sous leur version implantée dans Weka (Hall et al. (2009)) :

— L'algorithme bayésien utilisé est NaiveBayes [1].
— L'algorithme de classification par arbre de décision utilisé est C4.5 [2].

1. http ://weka.sourceforge.net/doc/weka/classifiers/bayes/NaiveBayes.html
2. http ://weka.sourceforge.net/doc/weka/classifiers/trees/J48.html

— L'algorithme à base de SVM appliqué est SMO (Sequential minimal optimization) [3].

Les algorithmes sont utilisés avec les paramètres par défaut de Weka. Les résultats présentés sont issus d'une validation croisée (10-fold). Notons qu'outre la validation croisée utilisée pour l'apprentissage du modèle de classification, dans nos futurs travaux, il sera nécessaire d'apprendre/tester les seuils de généralisation en utilisant un protocole de validation croisée similaire.

Chaque fonction de rang fournit des valeurs dans un espace différent. Pour permettre une juste comparaison, nous avons normalisé toutes les fonctions de façon à ce qu'elles renvoient des valeurs comprises entre 0 et 1.

5.2 Résultats

5.2.1 GENDESC

Les différentes fonctions afin de choisir les mots à généraliser ont été expérimentées et sont présentées dans cette section. Notons que la qualité d'une fonction ne dépend que très peu de l'algorithme d'apprentissage. Ce point sera discuté en Section 5.2.3.

La Table 5 montre les résultats obtenus en utilisant l'algorithme NaivesBayes, sans utilisation de n-grammes, avec différentes fonctions et plusieurs seuils de généralisation. Dans le premier corpus DEFT2007, la classification fondée sur les mots retourne un taux d'exactitude de **60,41%**. Dans le second corpus (TWEET) pour lequel il existe cinq classes distinctes, une classification fondée sur les mots retourne un taux d'exactitude de **46,80%**. Ces résultats constituent les valeurs de base d'une représentation classique "sac de mots".

Corpus	**DEFT2007**				**TWEET**			
Seuil	0,1	0,3	0,5	0,7	0,1	0,3	0,5	0,7
D	**63,49**	**68,26**	**63,18**	58,82	46,80	**50,24**	**49,19**	**47,38**
DF	59,23	58,72	58,92	59,74	46,32	45,44	43,90	43,70
IDF	59,26	54,56	53,75	54,67	**47,47**	41,95	26,85	25,10
TF	53,55	53,65	53,75	53,75	19,62	19,62	19,62	19,62
$DF \times D$	**60,95**	59,74	58,92	58,82	46,32	44,37	43,36	43,22
$D \times IDF$	**62,17**	53,85	53,85	53,85	**49,40**	35,50	20,13	20,13
$TF \times D$	**63,18**	**67,85**	**63,59**	59,84	46,80	46,26	43,22	45,37
$TF \times IDF$	55,48	54,57	53,65	53,55	39,66	28,19	22,80	21,95
$TF \times DF$	59,33	58,62	59,26	59,53	46,26	45,44	43,83	43,63

TAB. 5 – *Exactitude (Accuracy) obtenue en fonction des différentes fonctions selon des seuils.*

La Table 6 montre que **seule la fonction D est réellement pertinente pour établir quels mots doivent être généralisés**. Les autres fonctions fournissent globalement des résultats inférieurs à l'utilisation des mots comme descripteurs, quel que soit le seuil.

Sur le corpus TWEET, nous avons vu qu'il pouvait être intéressant, à condition d'utiliser un seuil faible, de combiner D avec IDF, alors qu'IDF avait plutôt tendance à dégrader les

3. http ://weka.sourceforge.net/doc/weka/classifiers/functions/SMO.html

performances de D sur le corpus parlementaire. Nous pensons que cela s'explique par le fait que le corpus TWEET est composé de cinq classes comparativement au corpus d'interventions parlementaires formé de deux classes.

Le seuil optimal varie suivant la fonction utilisée. Nous avons testé différents seuils pour chaque fonction pour étudier le comportement de nos résultats en fonction du seuil. Comme le montre la Table 6, pour la fonction D, le seuil optimal est établi autour de $0,3$ pour les deux corpus testés.

Seuil	0,1	0,3	0,5	0,7
DEFT2007	63,49	**68,26**	63,18	58,82
TWEET	46,80	**50,24**	49,19	47,38

TAB. 6 – *Exactitude obtenue avec la fonction D avec différents seuils.*

5.2.2 GENDESC associé aux n-grammes

Nous avons effectué des expérimentations sur ces mêmes corpus en prenant en compte les n-grammes de mots. La Table 7 montre les résultats obtenus avec l'algorithme NaiveBayes et la fonction D à son seuil optimal. Pour chaque colonne de cette Table, nous avons : (1) une génération partielle des descripteurs avec GENDESC, (2) l'utilisation des mots comme descripteurs, (3) l'utilisation d'un filtre. Dans ce cas, les mots ayant un score inférieur au seuil donné ne sont pas généralisés mais supprimés.

Les résultats montrent que l'utilisation simultanée de bigrammes et trigrammes à la place des unigrammes a tendance à donner un score inférieur à l'utilisation des unigrammes seuls. L'utilisation de bigrammes et/ou trigrammes combinés à l'utilisation des unigrammes donne souvent un meilleur résultat que l'utilisation d'unigrammes seuls. Notons que cette amélioration apportée par les n-grammes est supérieure avec les n-grammes construits à partir de descripteurs issus de la généralisation partielle (colonne GENDESC vs. colonne "mots" de la Table 7). Cela laisse penser que la combinaison des n-grammes et de GENDESC est tout à fait pertinente.

La Table 7 montre également l'utilisation de la fonction D en tant que filtre afin d'éliminer les mots en dessous du seuil établi. Nous pouvons constater que ceci influe positivement sur la qualité de la classification en comparaison de l'utilisation des mots comme seuls descripteurs. Nos fonctions de sélections sont donc parfaitement pertinentes. Cependant, bien que cette amélioration soit significative, elle ne surpasse jamais l'utilisation de GENDESC qui semble l'approche la plus pertinente.

5.2.3 GENDESC et algorithmes d'apprentissage

Nous avons enfin effectué des expérimentations avec plusieurs algorithmes d'apprentissage afin de pouvoir les comparer. La Table 8 montre les résultats obtenus (1) à partir de GENDESC avec un seuil optimal, (2) sans l'utilisation de GENDESC.

Alors que NaiveBayes et SVM ont des performances relativement proches, l'algorithme s'appuyant sur des arbres de décision a des performances moindres, que ce soit en utilisant les mots comme descripteurs ou ceux issus de la généralisation partielle. Nous pouvons constater

Type de représentation	DEFT2007			TWEET		
	GENDESC	mots	Filtre	GENDESC	mots	Filtre
Unigrammes	68,26	57,25	66,29	51,67	46,80	51,38
Bigrammes	62,78	51,94	62,62	28,12	37,15	30,72
Trigrammes	58,11	45,34	61,44	20,40	25,76	20,59
Unigrammes + Bigrammes	67,55	56,61	66,79	52,48	47,74	51,52
Unigrammes + Trigrammes	**69,67**	57,12	66,92	**53,28**	47,00	51,45
Unig + Big + Trig	**68,36**	56,74	66,92	**53,36**	47,94	51,59

TAB. 7 – *Exactitude obtenue grâce aux n-grammes.*

que la qualité des descripteurs ne varie pas en fonction des algorithmes d'apprentissage : les descripteurs les plus pertinents le sont quel que soit l'algorithme d'apprentissage utilisé.

6 Conclusions et perspectives

Dans cet article, nous avons proposé de nouveaux descripteurs textuels fondés sur un mélange de mots et de POS-tag qui permettent d'améliorer la classification de documents par des méthodes d'apprentissage supervisé. Nos résultats montrent que notre approche est adaptée dans le cadre de la problématique de classification, sans connaissance *a priori* du corpus ou du type de classification à effectuer.

Alors qu'une part importante de notre approche était construite à partir de n-grammes, nous avons relevé que l'amélioration qu'ils donnaient était limitée. La faiblesse des n-grammes vient du fait qu'ils sont construits à base de mots voisins et non de mots en relation syntaxique. Nous expérimentons actuellement la généralisation partielle en construisant des "n-grammes" fondés sur les relations syntaxiques plutôt que sur les relations de voisinage.

Les étiquettes, et surtout les n-grammes d'étiquettes, nécessitent d'avoir les étiquettes les plus générales possibles. Nous allons proposer des solutions pour répondre à une telle problématique, entre autre en proposant plusieurs niveaux de généralisation, comme dans les travaux de Raymond et Fayolle (2010). Parallèlement, nous souhaitons utiliser nos méthodes sur d'autres corpus, pour tester l'adaptation de GENDESC à un contexte multilingue.

Références

Aggarwal, C. et C. Zhai (2012). A survey of text classification algorithms. In *Springer US*, pp. 163–222.

Béchet, N. (2009). *Extraction et regroupement de descripteurs morpho-syntaxiques pour des processus de Fouille de Textes*. These, Universite Montpellier II.

Béchet, N., M. Roche, et J. Chauché (2008). Explsa : utilisation d'informations syntaxico-sémantiques associées à lsa pour améliorer les méthodes de classification conceptuelle. In F. Guillet et B. Trousse (Eds.), *Acte d'EGC*, pp. 589–600.

Généralisation partielle avec GENDESC **:**

Types de combinaisons		**u**	**u+b**	**b**	**u+t**	**t**	**u+b+t**	**b+t**
DEFT2007	SVM	67,65	64,40	60,75	**68,46**	59,26	**65,52**	61,46
	Bayes	68,26	67,55	62,78	**69,67**	58,11	**68,36**	61,66
	Tree	59,84	59,74	55,88	**60,95**	52,23	**60,85**	55,68
TWEET	SVM	55,30	59,33	28,26	**59,53**	20,40	**59,80**	32,89
	Bayes	51,67	52,48	28,12	**53,29**	20,40	**53,36**	32,95
	Tree	38,52	39,13	20,13	**38,86**	20,13	**39,13**	20,13

Sans généralisation :

Types de combinaisons		**u**	**u+b**	**b**	**u+t**	**t**	**u+b+t**	**b+t**
DEFT2007	SVM	61,36	58,62	57,00	**62,98**	57,81	**59,63**	57,81
	Bayes	60,14	60,24	59,26	**60,95**	57,99	**60,55**	59,53
	Tree	55,38	54,77	52,13	**57,61**	54,16	**52,43**	55,38
TWEET	SVM	52,33	**54,89**	38,98	**52,60**	26,64	**54,69**	42,94
	Bayes	46,80	**47,74**	37,15	**47,00**	25,76	**47,94**	38,30
	Tree	43,43	**43,36**	21,98	**43,16**	19,96	**43,49**	21,78

TAB. 8 – *Résultats obtenus avec les différents algorithmes d'apprentissage.*
***u** indique la présence des unigrammes,*
***b** indique la présence des bigrammes,*
***t** indique la présence des trigrammes*

Chauché, J. (1984). Un outil multidimensionnel de l'analyse du discours. In *ACL*, ACL '84, pp. 11–15. Association for Computational Linguistics.

De Melo, G. et S. Siersdorfer (2007). Multilingual text classification using ontologies. In *Proceedings of the 29th European conference on IR research*, ECIR'07, pp. 541–548.

Fürnkranz, J. (1998). A study using n-gram features for text categorization. Technical Report OEFAI-TR-98-30, Austrian Research Institute for Artificial Intelligence.

Gamon, M. (2004). Sentiment classification on customer feedback data : noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of COLING '04*.

Grouin, C. (2008). Certification and cleaning up of a text corpus : Towards an evaluation of the "grammatical" quality of a corpus. In *Proceedings of LREC*.

Grouin, C., J.-B. Berthelin, S. E. Ayari, T. Heitz, M. Hurault-Plantet, M. Jardino, Z. Khalis, et M. Lastes (2007). Présentation de deft'07. In *Actes de l'atelier DEFT '07*, pp. 1–8.

Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, et I. H. Witten (2009). The weka data mining software : an update. *SIGKDD Explor. Newsl. 11*(1), 10–18.

Joshi, M. et C. Penstein-Rosé (2009). Generalizing dependency features for opinion mining. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pp. 313–316.

Landauer, T. K. et S. T. Dumais (1997). A solution to plato ?s problem : The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 211–240.

Nasukawa, T. et J. Yi (2003). Sentiment analysis : Capturing favorability using natural language processing. In *Proceedings of K-CAP '03*, pp. 70–77.

Plantié, M., M. Roche, G. Dray, et P. Poncelet (2008). Is a voting approach accurate for opinion mining ? In *Proceedings of DaWaK, LNCS, Springer Verlag*, pp. 413–422.

Porter, M. (1980). An algorithm for suffix stripping. *Program 14*(3), 130–137.

Porter, M. F. (1997). Readings in information retrieval. Chapter An algorithm for suffix stripping, pp. 313–316.

Pratt, F. (1996). *Secret and Urgent : The Story of Codes and Ciphers*. Cryptographic series. Français

Raymond, C. et J. Fayolle (2010). Reconnaissance robuste d'entités nommées sur de la parole transcrite automatiquement. In *TALN'10*.

Roche, M. et Y. Kodratoff (2003). Utilisation de LSA comme première étape pour la classification des termes d'un corpus spécialisé. In *Actes de la conférence MAJECSTIC'03*.

Salton, G. et M. J. McGill (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal 27*, 379–423, 623–656.

Tisserant, G., V. Prince, et M. Roche (2013). GenDesc : A Partial Generalization of Linguistic Features For Text Classification. In *Proceedings of NLDB'2013*, pp. 343–348.

Witten, I. H. et E. Frank (2005). *Data Mining : Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*.

Xia, R. et C. Zong (2010). Exploring the use of word relation features for sentiment classification. In *Proceedings of COLING '10*, pp. 1336–1344.

Summary

This paper presents an application that belongs to automatic classification of textual data by supervised learning algorithms. The aim is to study how a better textual data representation can improve the quality of classification. Considering that a word meaning depends on its context, we propose to use features that give important information about word contexts. We present a method named GENDESC, which generalizes (with POS tags) the least relevant words for the classification task.