Participation d'Orange Labs à DEFT2017

Aleksandra Guerraz Nicolas Voisine
Orange Labs
2, avenue Pierre Marzin, 22307 Lannion Cedex, France
{aleksandra.guerraz,nicolas.voisine}@orange.com

RÉSUMÉ
Cet article décrit la participation d'Orange Labs au DÉfi Fouille de Textes 2017 (DEFT2017). Nous détaillons d'abord notre contribution aux trois tâches proposées par le défi qui sont des problèmes de classification : classification des tweets selon leur polarité en *positive, négative, objective, mixte* pour les tâches 1 et 3, et classification qui consiste à identifier si le tweet contient ou non du langage figuratif pour la tâche 2. Pour ces trois tâches, nous proposons une méthode qui construit automatiquement les variables à partir de données multi-tables et utilise un classifieur bayésien naïf.

ABSTRACT
Orange Labs Participation at DEFT2017.

This article presents the participation of Orange Labs at DEFT2017. We decsribe our contribution to the three tasks proposed by the challenge which are classification tasks. The goal of tasks 1 and 3 is to classify the polarity of tweets into four classes (*positive, negative, objective, mixed*). Task 2 consists to identify if a tweet contains figurative expressions or not. For the three tasks we apply a method that automatically constructs variables from multiple tables data and uses a naive Bayes classifier.

MOTS-CLÉS : classification bayésienne naïve, données multi-tables, co-clustering, Khiops.

KEYWORDS: naive Bayes classifier, multiple tables data, co-clustering, Khiops.

1 Introduction

Le DÉfi Fouille de Textes 2017 (Benamara *et al.*, 2017) propose trois tâches de classification centrées sur l'analyse d'opinion et la détection du langage figuratif dans un corpus de tweets. Les tâches 1 et 3 sont des tâches de classification de tweets selon leur polarité. La tâche 2 consiste à identifier si le tweet contient ou non du langage figuratif.

Notre objectif est de tester dans ce contexte une méthode qui permet de faire de l'agrégation de données en construisant automatiquement un ensemble de variables à partir de données structurées en table.

Pour les trois tâches, nous utilisons Khiops un outil de préparation de données, de construction automatique de variables et de modélisation pour l'apprentissage supervisé (Boullé, 2016). Cet outil, développé à Orange Labs, est basé sur une méthode de classification qui utilise un classifieur bayésien naïf. Il est utilisé en interne dans de nombreux domaines applicatifs : marketing client, fouille du web, réseaux sociaux, caractérisation du trafic internet. Ici, nous l'appliquons à la fouille de textes.

Nous utilisons également une ressource externe sous forme de lexique de sentiments et d'émotions

constituée par (Abdaoui *et al.*, 2015) qui permet d'associer aux mots une polarité. Après une brève description des trois tâches dans la section 2, nous présentons notre approche dans la section 3. Nous terminons enfin par une discussion sur les résultats et une conclusion sur notre participation au défi.

2 Description des tâches

Le défi est divisé en trois tâches. Les tâches 1 et 3 consistent à classer les tweets selon leur polarité en quatre classes : *positive, négative, objective* et *mixte*. La classe *mixte* représente les tweets qui contiennent à la fois des opinions positives et négatives. A l'inverse de la tâche 1, les tweets de la tâche 3 peuvent utiliser le langage figuratif de type « ironie » ou « sarcasme », en excluant les tweets humoristiques. La distibution des classes dans les données d'apprentissage pour la tâche 1 est la suivante : 12% (*positive*), 32% (*négative*), 42% (*objective*) et 13% (*mixte*). Pour la tâche 3, la distribution des classes est : 9% (*positive*), 44% (*négative*), 33% (*objective*) et 12% (*mixte*). La distribution des classes pour ces deux tâches n'est donc pas très équilibrée.

La tâche 2 consiste à identifier si les tweets comportent du langage figuratif ou non. Trois types de langage figuratif sont considérés : l'ironie, le sarcasme et l'humour. La distribution des classes, pour cette tâche, est la suivante : 33% (*figurative*) et 67%(*nonfigurative*).

3 Méthode

Pour répondre aux trois tâches du défi, nous avons décidé de tester une seule méthode et de ne proposer qu'un essai par tâche. Nous avons mis au point notre méthodologie sur les données de la tâche 1 et nous l'avons appliquée sur l'ensemble des tâches proposées.

Ainsi, nous avons utilisé l'outil Khiops dont les principales fonctionnalités sont les suivantes :
- prise en compte des schémas multi-tables en étoile, avec une table racine comportant les individus (ici textes) et des tables secondaires en relation 0-1 ou 0-n contenant des enregistrements complétant les individus,
- construction automatique de variables pour créer une table $individus \times variables$,
- préparation des données supervisées par discrétisation et groupement de valeurs (Boullé, 2006),
- modélisation par classifieur bayésien naïf, avec pré-traitements univariés, sélection et moyennage de modèles.

Dans notre cas, les individus à analyser sont les tweets représentés sous forme d'identifiants et contenus dans une table racine. Leur description est complétée par des enregistrements contenus dans une table secondaire en relation 0-n. En effet, chaque tweet est représenté par l'ensemble des mots qui le composent. A chaque mot une description est associée (polarité, position du mot dans le tweet, etc...). L'ensemble des pré-traitements appliqués pour décrire les tweets sont détaillés dans la section suivante.

3.1 Pré-traitements

Pour tenir compte des particularités linguistiques des tweets, nous avons appliqué un certains nombre de pré-traitements. Tout d'abord, nous avons lemmatisé les tweets avec l'outil TreeTagger [1], ce qui a permis de normaliser un certains nombres d'unités comme par exemple les URL et les adjectifs numéraux.

Pour contextualiser les tweets, nous avons réalisé un co-clustering qui permet de partitionner simultanément deux variables à l'aide de Khiops Coclustering, comme proposé dans (Collin *et al.*, 2013). Les deux dimensions que nous utilisons sont : $Tweet \times Mot$. Le co-clustering est réalisé sur l'ensemble du corpus de chacune des tâches, à savoir sur les données d'apprentissage et de test. Nous utilisons ensuite les clusters de tweets pour la modélisation supervisée. Les tweets ne sont plus représentés uniquement par un ensemble de mots qui les composent, mais sont également classés dans un des clusters.

De plus, le rang de chaque mot (de 0 à 1) a été calculé et indique sa position relative dans le texte.

Pour associer aux mots une polarité, nous utilisons le lexique d'émotions et sentiments « FEEL » (Abdaoui *et al.*, 2015). Dans ce lexique, une polarité (positive ou négative) est attribuée à chaque terme et éventuellement une émotion parmi les six émotions suivantes : joie, colère, peur, surprise, dégoût et tristesse. Nous retenons seulement la polarité des mots qui dans le lexique sont marqués par une émotion. Les mots tels que *rouler*, *blanc* qui sont étiquetés initialement dans le lexique comme étant respectivement négatif et positif, sans être porteur d'aucune émotion, sont considérés comme étant neutres (nous ne leur attribuons aucune polarité). Nous avons également complété ce lexique par une centaine d'entrées suite à une analyse sur la distribution des mots dans les différentes classes. Des mots comme *bravo*, *satisfait*, *réjouir* ont été ajoutés avec la polarité positive et des mots tels que *acharnement*, *racisme* ont été ajoutés avec la polarité négative. De plus, nous avons complété le lexique avec une quarantaine d'émoticônes, les plus fréquentes, sous forme de leur code hexadécimal.

Pour prendre en compte la négation, nous avons mis en place une règle simple qui consiste à changer la polarité d'un mot s'il apparaît dans la structure syntaxique « ne...pas ».

En ce qui concerne le traitement des mots vides, nous nous limitons à supprimer les articles définis et indéfinis. En revanche, nous gardons les mots du type conjonction, pronoms etc... pour préserver des parties de discours pouvant être caractéristiques d'une opinion. Par exemple, la présence de la conjonction « mais » peut être un indicateur, pour la classe *mixte*, d'opposition ou de transition entre une opinion négative et une opinion positive :

– saison très très bof **mais** on se marrait
– franchement dals ça me saoule **mais** dommage pour l'élimination de olivier mine que j'aime beaucoup.

De même, nous ne filtrons pas les signes de ponctuation qui sont un moyen d'expression d'une émotion ou d'un sentiment qu'il soit positif ou négatif (étonnement, joie, colère, etc...) et surtout quand ils sont répétés. Voici un exemple de tweet positif et un exemple de tweet négatif utilisant des points d'exclamation :

– L'Equipe de France m'a appris que jamais rien n'est perdu ! On verra le match retour ! ! ! !
– J'ai honte de la France si je pouvais allez aux USA j'aurais un vrai président pas un clown sans personnalité. Ras le bol Sarkozy ! ! ! ! ! ! ! ! ! !

1. http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

Au final, nous mettons les données au format multi-tables. Pour la tâche 1, la table principale comporte 3906 instances avec trois variables : l'identifiant du tweet, l'identifiant du cluster dans lequel est mis le tweet après le déploiement du co-clustering et enfin la classe à prédire. La table secondaire, en lien 0-n avec la table principale, comporte l'identifiant du tweet, les mots lemmatisés, la position du mot dans le tweet et la polarité associée aux mots. La figure 1 représente les fichiers des données d'apprentissage avec la table principale (à gauche) et la table secondaire (à droite).

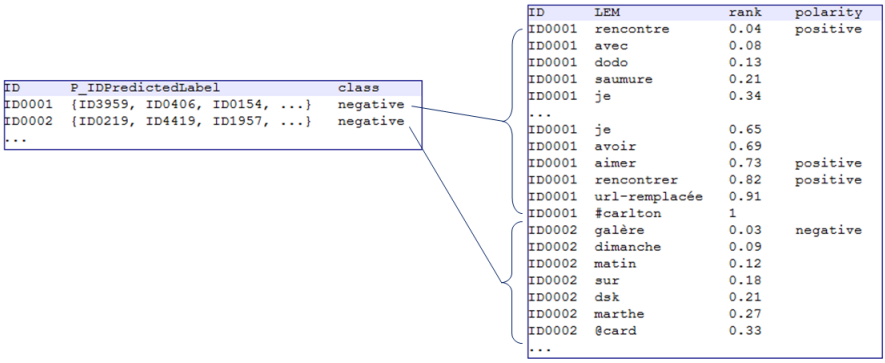

FIGURE 1 – Fichiers des données d'apprentissage pour la tâche 1

3.2 Apprentissage

Dans cette section, nous décrivons le modèle que nous avons obtenu avec Khiops pour la tâche 1. L'outil étant automatique, il est utilisable avec un minimum de paramétrage, ce qui constitue son point fort. Il suffit de spécifier les objectifs de l'analyse en précisant quelle est la variable à prédire et en mettant en entrée un dictionnaire qui décrit le schéma des données ainsi que les fichiers de données d'apprentissage et de test. Le paramètre utilisateur à définir est le nombre de variables à construire. Nous faisons varier ce paramètre pour examiner l'évolution des performances du modèle en termes d'AUC et de précision (de proportion d'instances correctement classées).

Pour la tâche 1, plus de cent mille variables ont été construites automatiquement et 3466 variables informatives ont été sélectionnées pour la modélisation.

Les variables construites automatiquement tiennent compte du nombre de lemmes dans le tweet, de la polarité des mots, de la présence ou absence de certains mots dans le tweet, de la position des mots dans le tweet. Les variables les plus complexes permettent d'agréger automatiquement ces différentes caractéristiques. Par exemple, la présence d'une URL en fin de tweet est fortement corrélée avec la classe *objective*, ce qui est illustré sur la figure 2. La variable construite a pour nom *CountDistinct(lems.LEM) where LEM = url-remplacée and rank > 0.77*. Elle peut s'interpréter comme le nombre d'URL en fin de tweet. La discrétisation associée, en deux intervalles, montre que lorsqu'une URL est présente (intervalle de droite) la classe *objective* est largement majoritaire.

Parmi les variables les plus informatives du modèle, nous trouvons également le cluster de tweets dans lequel est placé le tweet après le déploiement du co-clustering. La figure 3 présente la distribution des classes cibles dans les regroupements de clusters de tweets. La spécialisation en polarité de certains

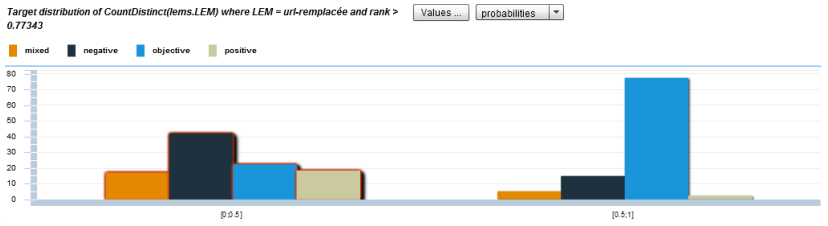

FIGURE 2 – Distribution des classes pour la variable « présence d'une URL »

clusters est nettement visible. Par exemple, la classe *objective* (en bleu) est largement majoritaire pour les deux premiers regroupements de clusters. La classe *négative* (en noir) est absente du septième groupement et est majoritaire pour le troisième groupement. De même, le troisième et septième regroupement comportent des tweets majoritairement associés à la classe *positive* (en beige).

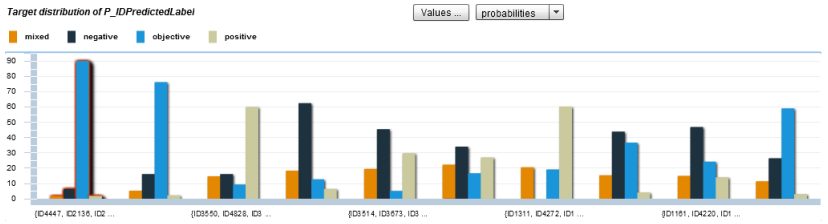

FIGURE 3 – Distribution des classes dans les clusters de tweets

La variable *Count(lems) where LEM not in {url-remplacée,être,.,avoir,pas,ne, !} and polarity positive*, représentée sur la figure 4 peut être interprétée comme le nombre de lemmes distincts autres que « URL, être, avoir, pas, ne, ! . » associés à une polarité positive. Quand ce nombre est suprérieur à deux, il est plus probable que le tweet appartienne à la classe *positive* (intervalle de droite). Si aucun des mots du tweet, hormis les mots cités ci-dessus, n'est associé à la polarité positive, il y a un mélange des quatre classes à prédire avec une nette sous-représentation de la classe *positive* (intervalle de gauche).

La présence d'un ou plusieurs points d'exclamation dans la deuxième partie du tweet est corrélée avec la classe *négative*, ce qui est illustré par la figure 5 (intervalle de droite).

De nombreuses variables intéressantes ont été construites. Celles-ci mettent en évidence les principales caractéristiques de nos données. Par exemple, les mots en fin de tweets marqués par une polarité ont un impact plus important sur polarité des tweets que les mots en début de tweet. Les hashtags représentant des titres d'émission, des noms propres peuvent être corrélés avec une opinion positive ou négative.

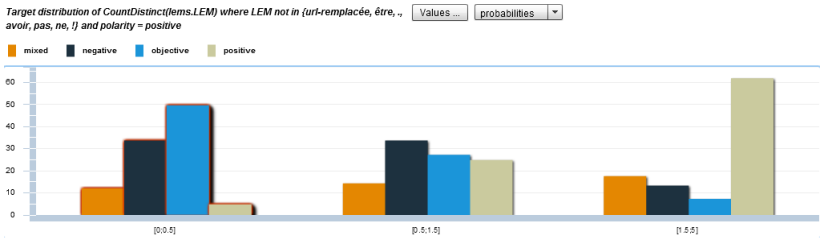

FIGURE 4 – Distribution des classes pour la variable qui tient compte du nombre de mots associés à une polarité *positive*

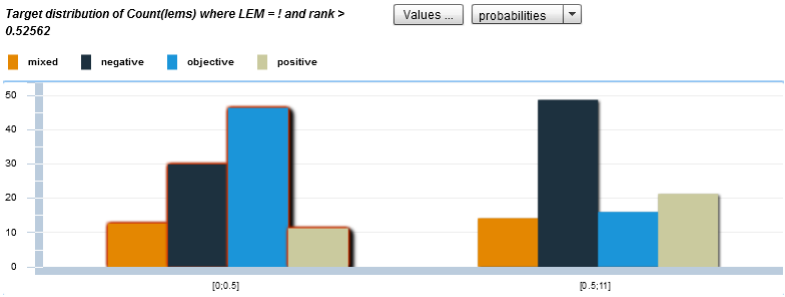

FIGURE 5 – Distribution des classes en fonction de la présence des points d'exclamation dans la deuxième partie du tweet

4 Résultats et discussion

Dans cette section, nous présentons les résultats obtenus pour les trois tâches. Tout d'abord, sur la figure 6 nous présentons l'évolution de la performance du modèle pour la tâche 1 en fonction du nombre de variables construites. La progression de la performance, à la fois de l'AUC (ligne rouge) et de la précision (points bleus), est visible jusqu'à dix mille variables construites. En revanche, à partir de ce nombre, la progression est moins nette et arrive sur un plateau entre dix mille et cent mille variables construites.

Pour évaluer l'apport de la polarité associée aux mots, nous avons construit des modèles pour lesquels nous n'avons pas recours au lexique de sentiments, en faisant également varier le nombre de variables. La figure 7, montre les différences de performances des modèles en termes d'AUC à gauche, et de précision à droite. L'utilisation de la polarité des mots permet d'augmenter significativement les scores à partir de cent variables construites.

Enfin, la table 1 présente les résultats officiels de notre participation en termes de macro précision, macro rappel et macro F-mesure. Ils montrent que le choix de la représentation des tweets que nous avons utilisée mérite d'être complétée. La possibilité d'intégrer différentes descriptions issues de sources diverses est particulièrement intéressante et simple à réaliser dans Khiops. Nous pourrons ainsi compléter la description des mots par leur représentation vectorielle, en utilisant, par exemple,

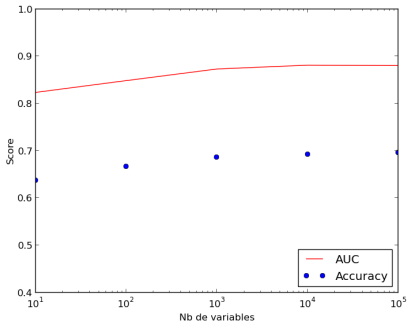

FIGURE 6 – Evolution de l'AUC et de la précision des modèles pour la tâche 1 en fonction du nombre de variables construites

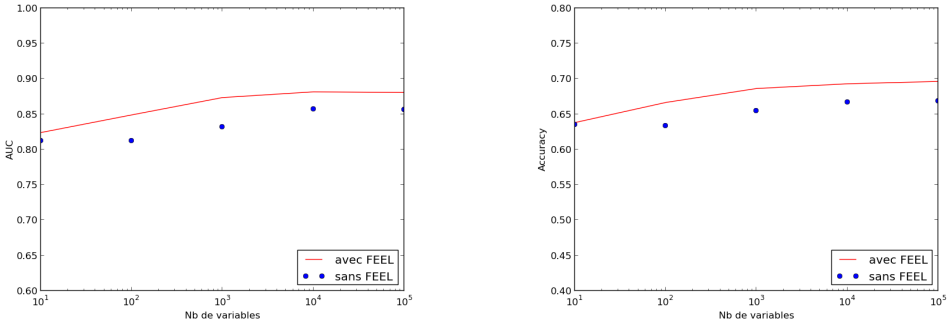

FIGURE 7 – AUC et précision des modèles en tenant compte (avec FEEL) ou pas (sans FEEL) de la polarité des mots

	Macro précision	Macro rappel	Macro F-mesure
Tâche 1	0,53	0,51	0,50
Tâche 2	0,66	0,66	0,66
Tâche 3	0,50	0,47	0,46

TABLE 1 – Résultats pour les trois tâches

word2vec, pour prendre en compte les similarités entre les mots à travers leurs contextes d'apparition.

5 Conclusion

Notre participation à DEFT2017 nous a permis de tester des outils développés à Orange Labs. La méthode que nous avons proposée constitue désormais une baseline, pour d'éventuels travaux futurs autour de la problématique d'analyse d'opinion et de sentiment. En effet, ce domaine intéresse de nombreuses entités de la Relation Client qui cherchent à comprendre les retours/commentaires des

clients sur les différents produits et services proposés par Orange. L'avantage majeur de notre méthode est de présenter une vision synthétique des données en créant des agrégats pertinents. Elle facilite l'analyse et la compréhension des données ce qui, dans un cadre opérationnel, constitue un atout.

En perspective de ce travail, nous comptons intégrer la détection de séquences qui permettra de capturer des régularités sur le plan syntaxique et stylistique dans les textes. Un travail sur la détection du langage figuratif, sur lequel nous ne nous sommes pas spécialement penchés, doit être également mené pour améliorer nos résultats (intégration de connaissances linguistiques, analyse des structures syntaxiques...).

Références

ABDAOUI A., NZALI M. D. T., AZÉ J., BRINGAY S., LAVERGNE C., MOLLEVI C. & PONCELET P. (2015). ADVANSE : Analyse du sentiment, de l'opinion et de l'émotion sur des Tweets Français. In *Actes de l'atelier DEFT de la conférence TALN*, Caen.

BENAMARA F., GROUIN C., KAROUI J., MORICEAU V. & ROBBA I. (2017). Analyse d'opinion et langage figuratif dans des tweets : présentation et résultats du Défi Fouille de Textes DEFT2017. In *Actes de l'atelier DEFT de la conférence TALN*, Orléans.

BOULLÉ M. (2016). Khiops : outil d'apprentissage supervisé automatique pour la fouille de grandes bases de données multi-tables. In *Actes de EGC 2016 (Extraction et Gestion des Connaissances)*, p. 505–510, Reims.

BOULLÉ M. (2006). MODL : a Bayes optimal discretization method for continuous attributes. *Machine Learning*, **65**(1), 131–165.

COLLIN O., GUERRAZ A., HIOU Y. & VOISINE N. (2013). Participation d'Orange Labs à DEFT 2013. In *Actes de l'atelier DEFT de la conférence TALN*, Les Sables-d'Olonnes.