Exploration de descripteurs de plongements de graphes pour la détection de messages abusifs

Noé Cécillon

AVIGNON UNIVERSITÉ

MÉMOIRE DE MASTER

Master Informatique

Parcours *Ingénierie du Logiciel pour la Société Numérique*

Centre d'Enseignement et de Recherche en Informatique

Laboratoire Informatique d'Avignon

Présenté par

Noé Cécillon

Exploration de descripteurs de plongements de graphes pour la détection de messages abusifs

---

Mémoire soutenu le 4 juillet 2019

Encadrement : Richard Dufour & Vincent Labatut (LIA)

LABORATOIRE INFORMATIQUE D'AVIGNON

Table des matières

1

Introduction

Le développement d'Internet a permis de mettre en relation des personnes du monde entier au travers de communautés en ligne qui regroupent des utilisateurs ayant des centres d'intérêts communs. Le nombre et la taille de ces communautés ne cessent de croître, ce qui leur confère une grande importance socio-économique. De nombreuses entreprises, de tous les domaines, sont ainsi intéressées par ce nouveau média et les opportunités d'échange qu'il offre. Chaque communauté a un règlement propre contenant des règles générales fondées sur le respect d'autrui et d'autres limites plus spécifiques au groupe en question. Une caractéristique des communautés en ligne est l'anonymat des utilisateurs qui peut parfois être le déclencheur de comportements abusifs, c'est-à-dire transgressant le règlement. Ce phénomène touche quasiment l'intégralité des communautés en ligne et il est primordial de le prendre au sérieux car il peut avoir un impact très important : ces comportements abusifs peuvent dégrader la qualité du service jusqu'à faire fuir une partie de la communauté. Dans les cas les plus extrêmes, des poursuites pénales peuvent même être engagées contre les administrateurs de ces plateformes. La modération est alors la procédure mise en place pour lutter contre ce phénomène en détectant les contenus et utilisateurs violant les règles de la communauté, et en appliquant des sanctions. Généralement cette tâche est effectuée manuellement par des modérateurs humains, ce qui rend ce processus fastidieux et coûteux.

Pour ces raisons, le développement de méthodes automatiques de modération suscite beaucoup d'intérêt. On peut distinguer deux approches : l'une, semi-automatique, qui va essayer de détecter les messages les plus susceptibles d'être abusifs afin de les porter à l'attention d'un modérateur humain et l'autre complètement automatique qui détecte toute seule les messages abusifs et applique les sanctions appropriées.

Dans cette étude, nous considérons cette tâche de modération automatique comme un problème de classification binaire consistant à déterminer automatiquement si un message est abusif ou non. Des travaux portant sur cette problématique ont déjà été effectués. On peut les décomposer en deux catégories : ceux utilisant le contenu des messages échangés et ceux utilisant leur contexte d'interaction entre utilisateurs, indépendemment du contenu échangé. Cette tâche n'est pas simple car nous travaillons sur des messages écrits en langage naturel très bruités, contenant alors potentiellement des fautes linguistiques, grammaticales, des abréviations... De même, il peut arriver que l'utilisateur choisisse intentionnellement d'utiliser un registre de langue permettant de contourner la surveillance des systèmes de modération automatiques actuels (par exemple, un système sera facilement capable

de détecter un mot “interdit” si celui-ci est bien orthographié, mais en sera incapable si l'utilisateur choisit de modifier l'ordre des lettres, d'en masquer certaines avec des “*”...). Dans ce travail, nous émettons l'hypothèse que le contenu des messages échangés et les interactions entre utilisateurs contiennent des informations différentes. Nous proposons une nouvelle méthode automatique de détection d'abus tirant parti de ces deux sources d'information. Pour ce faire, nous nous appuyons sur les travaux de Papegnies *et al*. qui ont développé une méthode utilisant le contenu des messages [9] ainsi qu'une autre méthode, ignorant complètement leur contenu textuel et s'appuyant uniquement sur une représentation de la conversation sous forme de graphes permettant de capturer les interactions entre utilisateurs [10]. Nous proposons trois stratégies capables de combiner ces approches et comparons leurs performances sur un corpus de messages provenant d'un jeu multijoueur en ligne. Nous effectuons ensuite une étude détaillée des descripteurs utilisés afin de trouver les plus importants dans le processus de classification. Le travail présenté ici correspond au travail exposé dans [3]. Notre contribution se divise en deux parties : 1) l'exploration des méthodes de fusion, et 2) l'analyse des descripteurs les plus discriminants pour ce problème.

Le reste de ce mémoire suit la structure suivante. Dans la Section 2, nous présentons les principaux travaux relatifs à la détection d'abus dans les messages textuels. Puis nous décrivons les approches utilisées dans cette étude dans la Section 3. Dans la Section 4, une description de nos données est proposée, ainsi que la discussion de nos résultats. Enfin, dans la Section 5, nous résumons nos contributions et présentons les perspectives pour ce travail.

2

État de l'art de la détection d'abus

De nombreux travaux de recherche ont été consacrés à la détection d'abus dans des messages textuels. La majorité d'entre eux se concentrent uniquement sur le contenu d'un message pour détecter son caractère abusif ou non. Spertus [13] propose une première tentative pour détecter des messages hostiles en se basant sur un ensemble de règles linguistiques prédéfinies. Dinakar *et al*. [6] décèle du harcèlement en ligne et Chen *et al*. [4] du langage offensant en utilisant des approches à base de n-grammes de mots. Dans le travail qui sert de base à notre étude, Papegnies *et al*. [9] proposent une méthode utilisant un grand nombre de descripteurs issus du contenu des messages (scores $tf\text{-}idf$, sac-de-mots, scores de sentiment, etc.). Récemment, des méthodes plus coûteuses en ressources ont été développées. Dans [14], Wulczyn *et al*. proposent trois ensembles de données contenant des messages provenant des pages de discussion de Wikipedia. Ces trois ensembles sont annotés respectivement pour des attaques personnelles, de l'agressivité et de la nuisibilité. Ces données ont été utilisées dans des travaux récents [8, 11] pour entraîner des réseaux de neurones récurrents opérant sur des plongements de mots (*word embeddings*) et des descripteurs à base de n-grammes. Ces approches traitant du contenu sont généralement peu coûteuses computationnellement et constituent une bonne référence en termes de performance. Cependant, la très grande variété d'utilisateurs, des langues, les fautes de frappe, les abréviations ou les obsfuscation volontaires de mots pour échapper aux filtres automatiques rendent quasiment impossible la création de modèles suffisamment robustes pour pouvoir être utilisés dans toutes les situations. Hosseini *et al*. [7] montrent même qu'il est très facile de contourner les systèmes automatiques de détection en rendant le contenu abusif difficile à déceler, notamment en remplaçant certaines lettres par d'autres caractères, en faisant volontairement des fautes d'orthographe ou en utilisant des sous-entendus. Par exemple "*connard*" peut être transformé en "*konar*" par certains utilisateurs ou "*con*" peut devenir "*c0n*".

Les messages abusifs provoquent souvent de vives réactions de la part des autres utilisateurs. C'est pour cela que certains auteurs considèrent des conversations complètes au lieu de n'utiliser que des messages individuels. Ainsi, Yin *et al*. [15] utilisent des descripteurs calculés à partir des phrases entourant le message à traiter. Balci et Salah [1] tirent parti d'informations sur les utilisateurs telles que le genre, le nombre d'amis, l'argent dépensé ou le temps passé sur la plateforme en ligne pour faire de la détection d'abus. Enfin, Papegnies *et al*. [10] proposent une méthode ignorant complètement le contenu textuel des messages et utilisant une représentation des conversations sous la forme d'un graphe conversationnel. Dans cette approche, ce sont les interactions entre les utilisateurs et les

évolutions dans la dynamique de la conversation qui permettent de détecter du contenu abusif. Dans le travail que nous proposons dans cet article, nous souhaitons alors améliorer la robustesse des systèmes de détection d'abus dans des conversations textuelles en tirant profit du contenu des messages mais également des interactions entre utilisateurs, tout en analysant la complémentarité entre ces sources d'information.

3

Méthodes

Dans cette section, nous présentons tout d'abord la méthode s'appuyant sur le contenu textuel des messages [9] (Section 3.1). Nous non intéressons ensuite à l'approche s'appuyant sur les interactions entre utilisateurs [10] (Section 3.2). Enfin, nous présentons les méthodes de fusion que nous proposons dans cet article, cherchant à tirer avantage de ces deux sources d'information. La Figure 3.1, représentant l'application complète, est présentée tout au long de cette section.

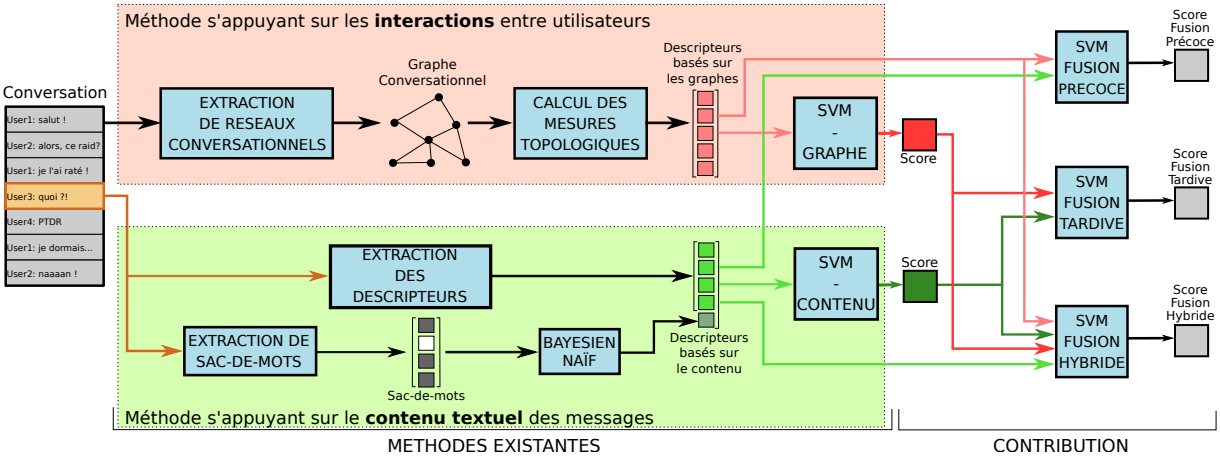

Figure 3.1 – Représentation de notre application complète. *Méthodes existantes* fait référence au travail décrit dans [9] (approche utilisant le contenu) et [10] (approche utilisant les interactions), tandis que la contribution de ce papier apparaît sur la droite (stratégies de fusion).

3.1 Contenu textuel

Cette méthode, que nous nommerons *Contenu* dans le reste du document, correspond à la partie inférieure gauche (partie verte) de la Figure 3.1. Elle consiste à extraire un certain nombre de descripteurs à partir du contenu du message que l'on veut classifier et d'entraîner une machine à vecteurs de support (SVM) afin de différencier les messages abusifs (classe *Abus*) et les messages non abusifs (classe *Non Abus*). Les descripteurs que nous exploitons sont assez standards, donc nous ne les décrivons que brièvement.

Dans un premier temps, nous utilisons des descripteurs morphologiques. Nous distinguons

6 classes de caractères différents (lettre, chiffre, ponctuation, majuscule, espace, et autre). Nous calculons 2 valeurs relatives à ces classes : le nombre d'occurrences pour chaque classe et le ratio de caractères faisant partie de cette classe par rapport au nombre total de caractères dans le message. Nous calculons également le nombre de caractères différents dans le message, la longueur du message, la taille du mot le plus long et la taille moyenne des mots du message. Toutes ces valeurs sont exprimées en nombre de caractères. De plus, les messages abusifs contiennent parfois un très grand nombre de *copier/coller*. Pour contrer ce phénomène, nous utilisons l'algorithme de compression Lempel–Ziv–Welch (LZW) [2] qui permet de compresser les parties d'un message qui sont répétées à de multiples reprises. Nous calculons un descripteur correspondant au ratio de la taille du message de base comparée à la taille du message compressé (avec des tailles exprimées en nombre de caractères). Enfin, les utilisateurs abusifs ont également tendance à utiliser des mots exagérément longs en répétant plusieurs fois des lettres. On crée une version comprimée du message dans lequel toutes les lettres répétées plus de deux fois consécutivement sont supprimées. Par exemple, "*mdrrrrrrrrrrrrrr*" serait transformé en "*mdrr*". On calcule la différence de longueur entre le message original et le message comprimé.

Nous avons également recours à des descripteurs de langage. Nous calculons le nombre de mots, le nombre de mots différents, et le nombre de mots nuisibles dans le message. Cette dernière valeur est calculée en utilisant une liste prédéfinie d'insultes, de symboles et de mots considérés comme étant représentatifs d'un abus. Nous avons manuellement constitué cette liste en recoupant des données disponibles en ligne. Ce descripteur est calculé pour le message de base ainsi que pour sa version comprimée. Nous calculons également deux scores $tf\text{-}idf$ correspondant aux sommes des scores $tf\text{-}idf$ standards de chacun des mots du message. Un score est calculé relativement à la classe *Abus* et l'autre relativement à la classe *Non Abus*. Ces scores sont également calculés sur la version comprimée du message. Enfin, nous appliquons une normalisation très simple au message en retirant les signes de ponctuation et en transformant toutes les lettres en minuscules afin de représenter notre message sous la forme d'un sac-de-mots (*Bag-of-Words*). Nous entraînons un classifieur Bayésien Naïf à détecter du contenu abusif en utilisant ces vecteurs binaires creux. Cette partie est représentée tout en bas de la Figure 3.1. La valeur de sortie de ce classifieur Bayésien Naïf est utilisée comme un descripteur pour notre système complet. Nous obtenons un total de 29 descripteurs (19 morphologiques et 10 liés au langage).

3.2 Modélisation des interactions entre utilisateurs

La méthode utilisant les interactions entre utilisateurs, nommée *Graphe* dans le suite du document, est représentée en haut à gauche de la Figure 3.1 (partie rouge). Elle ignore complètement le contenu textuel des messages et n'utilise que la dynamique de la conversation, en s'appuyant sur les échanges entre utilisateurs. Cette approche se décompose en 3 étapes :

1. Extraire un graphe conversationnel en se basant sur le message que l'on souhaite classifier et sur les messages le précédant et le suivant

2. Calculer des mesures topologiques sur ce graphe pour caractériser sa structure
3. Utiliser ces mesures comme des descripteurs pour entraîner un SVM permettant de distinguer les messages abusifs et non abusifs

Les nœuds de ce graphe représentent les utilisateurs prenant part à la conversation, et les liens pondérés décrivent l'intensité des interactions entre ces utilisateurs.

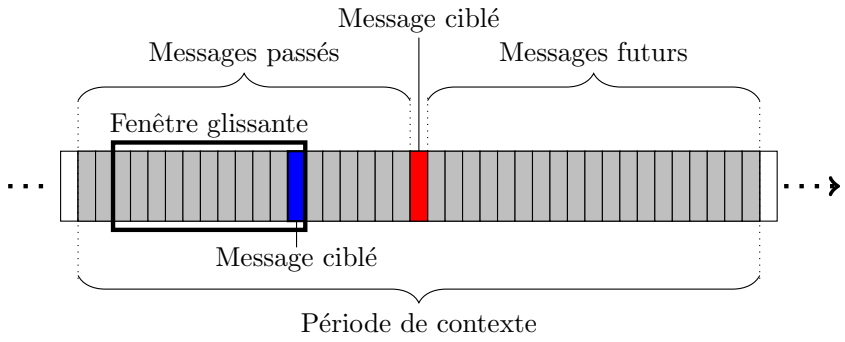

Figure 3.2 – Illustration des concepts principaux utilisés durant l'extraction du réseau conversationnel.

Les concepts utilisés pour extraire le graphe conversationnel sont représentés sur la Figure 3.2 dans laquelle chaque rectangle représente un message. Pour extraire le graphe, nous utilisons une *période de contexte* qui correspond à une séquence de messages centrée sur le *message ciblé*, *i.e.* que l'on souhaite classifier (représenté en rouge sur la Figure 3.2). Chaque utilisateur ayant posté au moins un message dans cette période est représenté par un nœud dans le graphe généré. Pour créer les liens et mettre à jour les poids qui leur sont associés, nous utilisons une *fenêtre glissante*, permettant alors de faire un zoom sur une partie de la conversation. Nous faisons glisser cette fenêtre sur l'intégralité de la période, selon un pas choisi (dans notre contexte, un pas de 1 message). A chaque itération, le graphe est mis à jour en ajoutant de nouveaux liens ou en ajustant leurs poids si ils existent déjà. A chaque instant, le dernier message de la fenêtre glissante (en bleu sur la Figure 3.2) est appelé *message courant* et son auteur *auteur courant*. La méthode de mise à jour des poids se base sur l'hypothèse que ce message est destiné aux autres utilisateurs présents dans la fenêtre. Des liens entre l'auteur courant et les utilisateurs dans la fenêtre sont créés ou, si ces liens existent déjà, leurs poids sont augmentés. De plus, on considère également que plus un auteur a posté récemment, plus il est probable que le message courant lui soit adressé. La chronologie est donc prise en compte dans le processus d'attribution des poids.

Lorsque la fenêtre a parcouru l'intégralité de la période de contexte, nous obtenons un graphe conversationnel que nous appelons le réseau *Complet*. Nous extrayons deux variantes de réseaux supplémentaires de plus petite taille, s'appuyant sur le même contexte : les réseaux *Avant* et *Après* basés respectivement sur les messages postés avant et après le message ciblé (en plus de ce message lui-même). La Figure 3.3 montre un exemple de ces 3 réseaux pour un message abusif.

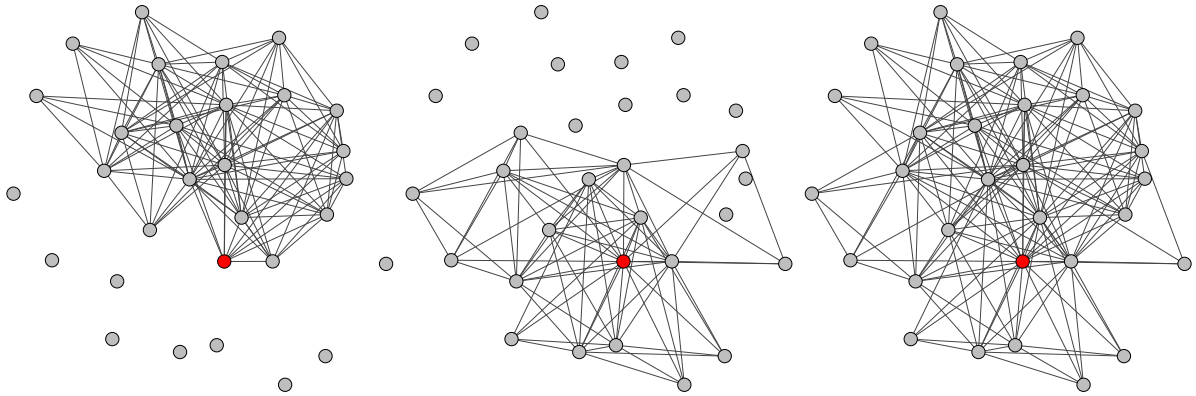

Figure 3.3 – Exemple de réseaux conversationnels extraits pour un contexte donné : *Avant* (gauche), *Après* (centre), et *Complet* (droite). L'auteur du message abusif est représenté en rouge.

Une fois les réseaux conversationnels extraits, ils doivent être représentés par des valeurs numériques afin de pouvoir être utilisés par le classifieur SVM. Nous faisons cela en calculant un ensemble de mesures topologiques standards qui permettent de caractériser le graphe de différentes manières en se focalisant sur différentes *échelles* et *portées*. L'échelle correspond à la nature de l'entité étudiée. Dans ce travail, nous considérons des mesures locales qui décrivent un nœud individuellement et des mesures globales qui prennent en compte l'intégralité du graphe. L'échelle peut être micro-, méso- ou macroscopique et correspond à la quantité d'information considérée par la mesure. Par exemple, la densité du graphe est microscopique, la modularité est mésoscopique et le diamètre est macroscopique. De plus, la majorité de ces mesures peut prendre en compte ou non la direction et le poids des liens. Ainsi, différentes variantes d'une même mesure sont extraites et utilisées par le classifieur. Toutes ces mesures sont calculées pour chacun des trois réseaux et sont ensuite utilisées comme descripteurs pour entraîner le SVM. Dans ce travail, nous utilisons exactement les mêmes 459 mesures que dans [10].

3.3 Fusion

On propose maintenant une nouvelle approche visant à combiner les 2 méthodes décrites précédemment. Cette méthode s'appuie sur l'hypothèse que les descripteurs extraits à partir du contenu et du contexte contiennent des informations différentes. Ces informations pourraient être complémentaires, et leur combinaison pourrait donc permettre d'améliorer les performances de classification. Nous testons trois stratégies de fusion différentes. Cette approche est représentée sur la droite de la Figure 3.2.

La première stratégie est la *Fusion Précoce*. Elle consiste à former un ensemble global de descripteurs, regroupant tous les descripteurs des approches utilisant le contenu et les graphes des Sections 3.1 et 3.2. Cet ensemble de descripteurs est ensuite utilisé pour entraîner un nouveau SVM. Le raisonnement sous-jacent à cette stratégie est que le classifieur a ainsi

accès à l'intégralité des descripteurs et qu'il peut donc déterminer lesquels sont importants pour traiter ce problème.

La seconde stratégie est la *Fusion Tardive*, qui se décompose en 2 étapes. Dans un premier temps, on applique séparément les deux méthodes des Sections 3.1 et 3.2. Pour chacune, on produit un score correspondant à la probabilité du *message ciblé* d'être abusif. On obtient donc deux scores qui sont utilisés comme descripteurs d'entrée pour un nouveau SVM. L'intuition est que ces scores pourraient permettre d'effectuer un filtrage des informations en conservant uniquement les plus importantes pour le nouveau classifieur.

Enfin, la troisième stratégie peut être considérée comme une *Fusion Hybride*. En effet, il s'agit d'une combinaison des 2 stratégies précédentes. On crée un ensemble contenant tous les descripteurs des approches utilisant le contenu et les graphes comme dans la *fusion précoce*, et on y ajoute les deux scores utilisés dans la *fusion tardive*. Ce nouvel ensemble de descripteurs est utilisé pour entraîner un SVM. L'idée est de vérifier si les scores permettent bien de regrouper toutes les informations importantes des descripteurs de base, et, si ce n'est pas le cas, le classifieur a accès à ces informations en utilisant directement les descripteurs de base.

4

Données et expérimentation

Dans cette section, nous présentons dans un premier temps les données que nous utilisons (Section 4.1), puis le protocole expérimental utilisé pour évaluer et valider nos approches (Section 4.2). Ensuite, nous présentons et discutons les résultats obtenus en termes de performance de classification (Section 4.3). Enfin, nous effectuons une analyse des performances obtenues (Section 4.4) afin de mettre en lumière les descripteurs importants dans les performances de classification.

4.1 Données

Pour ce travail, nous avons eu recours au même jeu de données que celui utilisé dans les travaux servant de base à cette étude [10, 9]. Ce jeu de données privé contient 4 029 343 messages écrits en français et provenant du système de messagerie instantanée intégré au jeu *SpaceOrigin* [1]. Il s'agit d'un jeu de rôle en ligne massivement multijoueur (MMORPG) français. Parmi ces messages, 779 ont été signalés par au moins un joueur et ensuite confirmés par un modérateur humain comme étant abusifs. Ils constituent notre classe *Abus*. Quelques inconsistances dans la base de données nous empêchent de récupérer le contexte de certains messages qui ne peuvent donc pas être utilisés. Après filtrage de ces messages, la classe *Abus* contient 655 messages. Pour avoir un corpus équilibré, nous extrayons aléatoirement le même nombre de messages parmi tous ceux n'ayant jamais été signalés. Ces messages constituent notre classe *Non abus*. Chaque message, indépendamment de sa classe, est associé à son contexte, c'est-à-dire aux messages appartenant à la même conversation.

4.2 Expériences

Les descripteurs de la méthode *Contenu* ont été extraits au moyen de la bibliothèque Scikit-Learn [12]. Pour la méthode *Graphe*, les descripteurs ont été extraits des graphes conversationnels grâce à la librairie iGraph [5]. Les classifieurs que nous utilisons sont ceux de Scikit-Learn implémentés sous le nom de SVC (C-Support Vector Classification). Au vu de la petite taille de notre corpus de données, nous effectuons nos expériences en

1. https ://play.spaceorigin.fr/

faisant une validation croisée sur 10 échantillons. Chaque échantillon est équilibré entre les classes *Abus* et *Non abus*. 70% des données sont utilisées pour l'entraînement et 30% pour le test. La méthode d'extraction des graphes employée dans la partie *Graphes* utilise 2

paramètres : la taille de la *période de contexte* et la taille de la *fenêtre glissante*. Cette dernière est fixée à une longueur de 10 messages par rapport à des contraintes ergonomiques de l'interface graphique du jeu dont nos données sont extraites. La *période de contexte* contient 1 350 messages. Ces valeurs sont celles obtenant les meilleures performances lors de l'étude faite dans [10].

4.3 Résultats

Méthode	**Nombre de descripteurs**	**Durée d'exécution**	**Durée moyenne**	**Précision**	**Rappel**	***F*-mesure**
Contenu	29	0 :52	0,02s	78,59	83,61	81,02
Contenu - TD	3	0 :21	0,01s	75,82	82,57	79,05
Graphes	459	8 :19 :10	7,56s	90,21	87,63	88,90
Graphes - TD	10	14 :22	0,03s	88,72	84,87	86,75
Fusion Précoce	488	8 :26 :41	7,68s	91,25	89,45	90,34
Fusion Précoce - TD	4	11 :29	0,17s	89,09	87,12	88,09
Fusion Tardive	488 (2)	8 :23 :57	7,64s	94,10	92,43	93,26
Fusion Tardive - TD	13	15 :42	0,24s	91,64	89,97	90,80
Fusion Hybride	490	8 :27 :01	7,68s	91,96	90,48	91,22
Fusion Hybride - TD	4	16 :57	0,26s	90,74	89,00	89,86

Table 4.1 – Comparaison des performances obtenues avec les méthodes *Contenu*, *Graphe*, *Fusion*, et leurs sous-ensembles de *Top Descripteurs* (TD) respectifs (présenté dans la Section 4.4). La durée d'exécution est exprimée en *heure* :*minute* :*seconde*.

Le Tableau 4.1 présente les résultats en termes de Précision, Rappel et *F*-Mesure obtenus sur la classe *Abus* pour les deux méthodes *baseline*, *Contenu* [9] et *Graphe* [10], ainsi que pour les 3 nouvelles stratégies de fusion proposées (*Fusion précoce*, *Fusion tardive* et *Fusion hybride*). Il présente également le nombre de descripteurs utilisés pour effectuer la classification, le temps total pour calculer les descripteurs et effectuer la validation croisée (*Durée d'exécution*), et le temps moyen pour traiter un message (*Durée moyenne*). Il est à noter que la *Fusion Hybride* n'utilise que 2 valeurs d'entrée (les scores des méthodes *Contenu* et *Graphe*) mais ces méthodes utilisent des valeurs d'entrée différentes, ce qui explique les valeurs indiquées dans le Tableau 4.1.

La première observation que l'on peut faire est que, indépendamment de la stratégie, les performances obtenues par les méthodes de fusion sont meilleures que celles de base (*Contenu* et *Graphe*). Cela confirme l'hypothèse que l'information présente dans la méthode utilisant le contenu est différente de celle contenue dans la méthode utilisant les graphes, ces 2 sources d'information étant au moins partiellement complémentaires puisque les performances de classification sont améliorées quand on les fusionne.

La seconde observation est que la *fusion tardive* est la stratégie de fusion qui obtient les meilleures performances en atteignant une *F*-Mesure de 93,26 %. C'est assez surprenant car, rappelons le, c'est la stratégie qui n'utilise que 2 descripteurs comme données d'entrée : les 2 scores de sortie des méthodes de référence. En comparaison, la *fusion précoce* a, elle, accès à un nombre beaucoup plus important de descripteurs (488). Il semble donc que les méthodes *Contenu* et *Graphe* soient efficaces pour condenser les données nécessaires à la classification dans un seul score de sortie et ne pas perdre d'information trop importante. On peut aussi supposer que la *fusion précoce* a des difficultés à trouver un modèle approprié à ce problème, probablement à cause d'un nombre trop élevé de descripteurs à traiter. La *fusion tardive* bénéficie elle d'une sorte de pré-traitement effectué par ses 2 prédécesseurs qui réduisent la dimension des données en produisant les 2 scores. Cela semble être confirmé par la *fusion hybride* qui obtient de meilleurs résultats que la *fusion précoce* mais reste en dessous de la *fusion tardive*. Cette piste pourrait être explorée plus en détail en utilisant un classifieur moins sensible au nombre de descripteurs à traiter.

Pour le temps de traitement, l'approche à base de graphes nécessite 8 heures et 19 minutes pour traiter l'intégralité de notre corpus. La phase d'extraction des descripteurs est responsable d'environ 95% de ce temps de traitement. Cela est dû au grand nombre de descripteurs à calculer (459) et à la complexité computationnelle pour calculer certains d'entre eux. Pour les raisons inverses, la méthode utilisant le contenu est beaucoup plus rapide avec un temps total d'exécution de moins d'une minute. En effet, cette méthode n'exploite qu'un nombre restreint de descripteurs (29), rapides à calculer. Les méthodes de fusion ont besoin de calculer les descripteurs de ces deux approches, ce sont donc elles qui ont les temps de traitement les plus longs.

De plus, comme indiqué précédemment, il existe deux types de systèmes automatiques de modération : les systèmes complètement automatisés et semi-automatiques. Pour adapter la méthode au besoin, il est possible de jouer sur la *précision* et le *rappel*, en privilégiant l'un par rapport à l'autre selon l'application visée. Ainsi, dans un système d'assistance semi-automatique, on va généralement essayer de maximiser le *rappel* afin de ne passer à côté d'aucun message abusif, quitte à détecter des messages qui ne devraient pas l'être, puisqu'un modérateur humain traitera ces alertes et prendra la décision finale. Au contraire, pour un système complètement automatisé, c'est la précision qui sera privilégiée pour ne pas sanctionner injustement des messages et des utilisateurs. En revanche, cela aura pour cause de ne pas détecter certains messages qui auraient pourtant dû l'être.

4.4 Étude des descripteurs

Maintenant que nous avons vu les performances obtenues par chaque approche, nous voulons savoir quels descripteurs sont les plus importants dans ce processus de classification. Pour cela, nous entraînons un estimateur de la bibliothèque *Scikit-Learn* sur nos données,

Méthode	Top descripteurs	Graphe	Poids	Directions	Échelle
Contenu	Score classifieur Bayésien naïf	–	–	–	–
	Score $tf–idf$ (Classe *Abus*)	–	–	–	–
	Ratio de majuscules	–	–	–	–
Graphes	Centralité coreness	C	–	E	G
	Centralité PageRank	Ap	N	O	N
	Force	C	P	S	N
	Nombre de sommets	C	–	–	G
	Centralité de proximité	Av	P	S	G
	Centralité de proximité	Av	P	S	N
	Score d'autorité	Av	P	O	G
	Score de Hub	Av	N	O	N
	Réciprocité	Ap	–	O	G
	Centralité de proximité	Ap	P	N	N
Fusion Précoce	Centralité coreness	Ap	–	S	G
	Centralité coreness	Av	–	E	G
	Excentricité	Av	–	E	G
	Score classifieur Bayésien naïf	–	–	–	–
Fusion Tardive	TD de *Contenu* ∪ TD de *Graphes*	–	–	–	–
Fusion Hybride	Score du SVM *Graphes*	–	–	–	–
	Score du SVM *Contenu*	–	–	–	–
	Force	Ap	P	S	N
	Centralité coreness	Av	–	E	G

Table 4.2 – Top descripteurs (TD) obtenus Pour nos 5 méthodes. Les lettres dans la colonne *Graphe* correspondent à *Avant* (Av), *Après* (Ap) et *Complet* (C). Celles dans les colonnes *Poids* et *Directions* correspondent à : *Non Pondéré* ou *Non Orienté* (N), *Pondéré* (P), *Orienté* (O), *Entrant* (E) et *Sortant* (S). Celles dans la colonne *Échelle* signifient *A l'échelle du graphe* (G) ou *A l'échelle du sommet* (N).

qui donne un classement des descripteurs d'entrée reflétant l'importance de chacun d'entre-eux pour la classification. En fonction de ce classement, nous appliquons une méthode itérative : le descripteur le moins discriminant est identifié, retiré et un nouveau classifieur est entraîné sur ce nouveau jeu de descripteurs. L'impact de cette suppression est mesuré par la différence de performance, en termes de F-Mesure. Ce processus est réitéré jusqu'à n'avoir plus qu'un seul descripteur. La Figure 4.1 montre l'évolution de la performance, en F-Mesure, au cours de ce processus appliqué à la méthode *Graphe*. Le sous-ensemble minimal de descripteurs permettant d'obtenir au moins 97 % de la performance originale, c'est-à-dire en utilisant l'ensemble des descripteurs, est appelé le sous-ensemble de *Top Descripteurs* (TD).

Cette méthode est appliquée aux deux *baselines* et aux trois stratégies de fusion afin d'obtenir les *Top descripteurs* de chacune de ces approches. Nous effectuons ensuite une classification pour chacune d'entre-elles en utilisant uniquement leurs *Top descripteurs* respectifs. Les résultats sont présentés dans le Tableau 4.1. Au niveau des performances, les méthodes sont classées dans le même ordre que lorsque l'on utilise tous les descripteurs. Pour la *fusion tardive*, le résultat présenté est obtenu en exploitant les scores générés par les SVMs

entraînés sur les *Top descripteurs* des approches *Contenu* et *Graphe*. Ces scores sont aussi ceux utilisés lors du calcul des *Top descripteurs* de la *fusion hybride* (en combinaison avec tous les descripteurs des deux méthodes *baseline*).

Figure 4.1 – Évolution de la performance en fonction du nombre de descripteurs supprimés pour la méthode *Graphe*.

Les *Top descripteurs* obtenus pour chacune des méthodes sont listés dans le Tableau 4.2, dont les quatre dernières colonnes présentent quelles variantes des descripteurs provenant de la méthode à base de graphe sont concernées. En effet, comme précisé dans la Section 3.2, la plupart des mesures topologiques peuvent utiliser ou non le poids ou la direction des liens ; elles peuvent être mesurées au niveau du graphe ou d'un noeud et peuvent être calculées sur chacun des 3 réseaux (*Complet*, *Avant*, *Après*).

Le sous-ensemble pour la méthode *Contenu* contient 3 *Top descripteurs*, à commencer par le score de prédiction du classifieur *Bayésien Naïf*. Ce n'est pas étonnant car il s'agit déjà de la sortie d'un classifieur complet, entraîné pour détecter du contenu abusif à partir de représentations sous forme de sac-de-mots. La deuxième est le *score* $tf\text{-}idf$ calculé par rapport à la classe *Abus*, ce qui prouve que considérer la fréquence d'apparition des termes permet d'améliorer les performances. Enfin, la troisième est le *ratio de majuscules* (proportion de lettres majuscules dans le message), qui est certainement causé par la tendance des utilisateurs abusifs à écrire leurs messages en majuscules pour les rendre plus visibles. Pour l'approche *Graphe*, les descripteurs les plus importants permettent de détecter des changements, au niveau du graphe conversationnel, dans le voisinage direct de l'auteur du message ciblé (*Centralité coreness*, *Force*), en termes de distance dans la centralité nodale moyenne au niveau du graphe complet (*Centralité de proximité*) et dans la réciprocité des échanges entre utilisateurs (*Réciprocité*). Pour cette approche, les *Top descripteurs* obtenus sont les mêmes que dans [10] et sont discutés plus en détails dans ce papier.

Pour la *Fusion précoce*, 4 *Top descripteurs* sont obtenus : le descripteur utilisant le score du classifieur *Bayésien Naïf* (*Contenu*) et 3 mesures topologiques (*Graphe*). Deux d'entre-eux sont des variantes de la *centralité coreness* de l'utilisateur ciblé, calculé à partir des graphes *Avant* et *Après*. La troisième mesure est son excentricité qui reflète des changements majeurs dans les interactions autour de l'utilisateur. Nous pouvons imputer cela à l'énervement causé par le message abusif concerné qui fait réagir rapidement et massivement les autres utilisateurs. Il y a également 4 *Top descripteurs* pour la *Fusion hybride* : les 2 scores obtenus à partir des SVMs ainsi que 2 mesures topologiques que l'on a déjà retrouvées dans d'autres sous-ensembles de *Top descripteurs*. Il y a la *Force* (aussi présente dans les *Top descripteurs* de *Graphe* et *Fusion tardive*) et la *Centralité coreness* (aussi présente pour *Graphe*, *Fusion précoce* et *Fusion tardive*). Pour la *Fusion tardive*, on obtient un total de 13 *Top descripteurs*. Les deux plus importants sont les scores des SVMs et les autres sont majoritairement des descripteurs qui sont aussi présents dans les autres sous-ensembles de *Top descripteurs*.

L'objectif premier de cette étude des descripteurs était d'avoir une meilleure compréhension du jeu de données et du processus de classification. On remarque ainsi que certains descripteurs sont (quasi-)inutiles pour la classification, et il n'est donc pas pertinent d'utiliser du temps de calcul pour extraire ces données. On note également que certaines mesures sont corrélées (notamment les différentes variantes d'une même mesure), et contiennent donc des informations semblables : il n'est pas nécessaire de conserver toutes ces informations en doublons. En général, pour un groupe de descripteurs corrélés, au maximum un seul descripteur de ce groupe est inclus dans les *Top descripteurs* (si l'information est utile pour la classification). Une autre possibilité intéressante de cette étude des descripteurs est de réduire le temps de calcul grâce à ces sous-ensembles de *Top descripteurs*. En effet, dans notre cas nous arrivons à conserver 97 % de la performance pour toutes nos méthodes en n'utilisant que quelques descripteurs importants contre des centaines dans le processus de base. Par exemple, la *fusion hybride* avec les *Top descripteurs* a besoin de seulement 3,5 % de la durée d'exécution de la *fusion hybride* classique et obtient tout de même 97 % de sa performance.

5

Conclusion et perspectives

Dans cet article, nous nous intéressons au problème de détection automatique de messages abusifs en ligne. Nous tirons profit de deux approches déjà proposées sur cette problématique, l'une utilisant le contenu textuel des messages [9] et l'autre les interactions entre utilisateurs [10], pour proposer une nouvelle méthode permettant d'utiliser simultanément ces deux sources d'information. Nous mettons en évidence que les descripteurs extraits dans les deux méthodes de base sont complémentaires et que leur combinaison permet d'améliorer les performances de classification pour atteindre une F-Mesure de 93,26 %. Une limite de notre travail est le coût computationnel élevé lié à l'extraction et au calcul de certaines mesures. Cependant, en menant une étude sur nos descripteurs, nous montrons qu'il est possible de considérablement réduire le temps de traitement nécessaire (moins de 4% du temps de base) tout en conservant au moins 97% de la performance initiale, en n'utilisant que le sous-ensemble des descripteurs les plus pertinents.

Une limite de ce travail concerne la taille réduite du jeu de données que nous utilisons pour mener cette étude. Une des perspectives est donc de travailler sur un jeu de données plus important afin de pouvoir tester nos méthodes à une échelle beaucoup plus grande. Cependant, les ensemble de données actuellement disponibles ne sont composés que de messages isolés alors que nous avons besoin de conversations complètes pour pouvoir utiliser et évaluer correctement nos méthodes. De plus, une autre perspective pour ce travail consisterait à tester ces approches sur des données dans une autre langue que le français. La méthode *Contenu* devrait être impactée négativement par ce changement ce qui ne devrait en revanche pas être le cas de l'approche *Graphe* puisqu'elle n'utilise pas du tout le contenu textuel des messages. De plus, il est probable que les comportements et réactions soient différents en fonction de la communauté étudiée. Il serait donc intéressant de pouvoir utiliser des données provenant d'une autre communauté, afin d'identifier si les observations que nous avons faites pour la communauté du jeu *Space Origin* s'appliquent aussi à d'autres communautés. Afin de répondre à ces limites, nous avons commencé à travailler sur un corpus de messages provenant des pages de discussion de Wikipedia proposé dans [14]. Il contient lui aussi des messages individuels mais nous pouvons reconstruire des conversations à partir des messages individuels et des informations contenus dans ce jeu de données. Ces messages étant en anglais, ce jeu de données nous permettra de traiter simultanément les deux possibilités évoquées précédemment (test à plus grande échelle et sur une langue différente du français).

Enfin, pour cette étude nous avons sélectionné manuellement quels descripteurs nous

voulions utiliser pour chacune de nos approches. Ce n'est pas forcément une stratégie optimale car, comme confirmé par l'étude des descripteurs menée dans la Section 4.4, un grand nombre de ces descripteurs est (quasi-)inutile dans le processus de classification. Une perspective pour cette étude serait d'automatiser la sélection de ces descripteurs en utilisant des techniques à base d'embeddings. Ceux-ci permettraient d'apprendre automatiquement une représentation de nos données textuelles et des graphes, au lieu d'utiliser des descripteurs pour les représenter comme nous l'avons fait dans ce travail.

Bibliographie

[1] K. Balci et A. A. Salah. "Automatic analysis and identification of verbal aggression and abusive behaviors for online social games". In : *Computers in Human Behavior* 53 (2015), p. 517-526. doi : 10.1016/j.chb.2014.10.025 (cf. p. 3).

[2] L. V. Batista et M. M. Meira. "Texture classification using the Lempel-Ziv-Welch algorithm". In : *Brazilian Symposium on Artificial Intelligence*. 2004, p. 444-453. doi : 10.1007/978-3-540-28645-5_45 (cf. p. 6).

[3] N. Cécillon, V. Labatut, R. Dufour et G. Linarès. "Abusive Language Detection in Online Conversations by Combining Content- and Graph-Based Features". In : *Frontiers in Big Data* 2 (2019), p. 8. doi : 10.3389/fdata.2019.00008. url : https://www.frontiersin.org/article/10.3389/fdata.2019.00008 (cf. p. 2).

[4] Y. Chen, Y. Zhou, S. Zhu et H. Xu. "Detecting offensive language in social media to protect adolescent online safety". In : *International Conference on Privacy, Security, Risk and Trust and International Conference on Social Computing*. IEEE, 2012, p. 71-80. doi : 10.1109/SocialCom-PASSAT.2012.55 (cf. p. 3).

[5] G. Csardi et T. Nepusz. "The igraph software package for complex network research". In : *InterJournal* 1695.Complex Systems (2006). url : http://igraph.sf.net (cf. p. 10).

[6] K. Dinakar, R. Reichart et H. Lieberman. "Modeling the detection of Textual Cyberbullying". In : *5th International AAAI Conference on Weblogs and Social Media / Workshop on the Social Mobile Web*. AAAI, 2011, p. 11-17. url : https://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/3841 (cf. p. 3).

[7] Hossein Hosseini, Sreeram Kannan, Baosen Zhang et Radha Poovendran. "Deceiving Google's Perspective API Built for Detecting Toxic Comments". In : *preprint arXiv :1702.08138* (2017) (cf. p. 3).

[8] Pushkar Mishra, Helen Yannakoudakis et Ekaterina Shutova. "Neural Character-based Composition Models for Abuse Detection". In : *CoRR* abs/1809.00378 (2018) (cf. p. 3).

[9] E. Papegnies, V. Labatut, R. Dufour et G. Linares. "Impact of content features for automatic online abuse detection". In : *International Conference on Computational Linguistics and Intelligent Text Processing*. T. 10762. Lecture Notes in Computer Science. Berlin, DE : Springer, 2017, p. 404-419. doi : 10.1007/978-3-319-77116-8_30 (cf. p. 2, 3, 5, 10, 11, 16).

[10] E. Papegnies, V. Labatut, R. Dufour et G. Linares. "Conversational Networks For Automatic Online Moderation". In : *IEEE Transactions on Computational Social Systems*. IEEE, 2019, p. 38-55. doi : 10.1109/TCSS.2018.2887240. url : https://ieeexplore.ieee.org/document/8629298 (cf. p. 2, 3, 5, 8, 10, 11, 14, 16).

[11] J. Pavlopoulos, P. Malakasiotis et I. Androutsopoulos. “Deep Learning for User Comment Moderation”. In : *1st Workshop on Abusive Language Online*. ACL, 2017, p. 25-35. url : http://www.aclweb.org/anthology/W/W17/W17-30.pdf (cf. p. 3).

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot et E. Duchesnay. “Scikit-learn : Machine learning in Python”. In : *Journal of Machine Learning Research* 12 (2011), p. 2825-2830. url : http://www.jmlr.org/papers/v12/pedregosa11a.html (cf. p. 10).

[13] E. Spertus. “Smokey : Automatic recognition of hostile messages”. In : *14th National Conference on Artificial Intelligence and 9th Conference on Innovative Applications of Artificial Intelligence*. AAAI, 1997, p. 1058-1065. url : http://dl.acm.org/citation.cfm?id=1867616 (cf. p. 3).

[14] E. Wulczyn, N. Thain et L. Dixon. “Ex Machina : Personal Attacks Seen at Scale”. In : *26th International Conference on World Wide Web*. 2017, p. 1391-1399. doi : 10.1145/3038912.3052591 (cf. p. 3, 16).

[15] D. Yin, Z. Xue, L. Hong, B. D. Davison, A. Kontostathis et L. Edwards. “Detection of harassment on Web 2.0”. In : *WWW Workshop : Content Analysis in the Web 2.0*. 2009, p. 1-7. url : http://www.cse.lehigh.edu/~brian/pubs/2009/CAW2/ (cf. p. 3).

Résumé

Ces dernières années, les réseaux sociaux ont permis aux utilisateurs du monde entier de se rencontrer et de discuter. Les administrateurs de ces plateformes en ligne se doivent d'empêcher les utilisateurs d'adopter des comportements inappropriés. Cette tâche de modération des échanges, principalement menée par des humains, est de plus en plus coûteuse et difficile à cause de la quantité toujours plus grande de données à traiter. Dans la cadre de messages textuels, des méthodes permettant d'automatiser cette tâche ont été proposées, s'appuyant principalement sur le contenu linguistique. Des travaux récents ont également montré que des descripteurs sur la structure de la conversation, sous la forme de graphes conversationnels, permettent de faciliter la détection des messages abusifs. Dans ce papier, nous proposons de tirer profit de ces deux sources d'information en proposant des stratégies de fusion combinant des descripteurs issus du contenu textuel et d'autres basés sur des graphes conversationnels. Nos expérimentations sur des messages de messagerie instantanée (*chat*) montrent que le contenu des messages mais aussi les interactions dans la conversation contiennent des informations partiellement complémentaires qui permettent d'améliorer les performances d'une tâche de classification de messages abusifs jusqu'à atteindre une F-Mesure de 93,26 %.

Abstract

In recent years, online social networks have allowed world-wide users to meet and discuss. As guarantors of these communities, the administrators of these platforms must prevent users from adopting inappropriate behaviors. This verification task, mainly done by humans, is more and more difficult due to the ever growing amount of messages to check. Methods have been proposed to automatize this moderation process, mainly by providing approaches based on the textual content of the exchanged messages. Recent work has also shown that characteristics derived from the structure of conversations, in the form of conversational graphs, can help detecting these abusive messages. In this paper, we propose to take advantage of both sources of information by proposing fusion methods integrating content- and graph-based features. Our experiments on raw chat logs show that the content of the messages, but also of their dynamics within a conversation contain partially complementary information, allowing performance improvements on an abusive message classification task with a final F-measure of 93.26%.

AVIGNON UNIVERSITÉ