Détection de messages abusifs au moyen de réseaux conversationnels

Etienne Papegnies, Richard Dufour, Vincent Labatut, Georges Linarès

Détection de messages abusifs au moyen de réseaux conversationnels

Étienne Papégnies , Richard Dufour , Vincent Labatut , Georges Linarès

Laboratoire Informatique d'Avignon – EA 4128

prenom.nom@univ-avignon.fr

RÉSUMÉ. *Le nombre et la taille des communautés en ligne ne cessent de s'accroître depuis l'apparition du Web. Cependant, la modération du contenu produit par leurs utilisateurs est toujours essentiellement réalisée manuellement. L'automatisation de cette tâche permettrait de réduire son coût financier, mais la majorité des approches utilisées en production s'appuient uniquement sur le contenu des messages et apparaissent très sensibles aux méthodes d'obfuscation intentionnelles. Dans cet article, nous proposons une méthode pour l'extraction de réseaux conversationnels à partir des logs de conversations (chat), et nous étudions le pouvoir discriminant d'un ensemble de mesures topologiques pour un problème de classification consistant à identifier les messages abusifs. Notre approche atteint un niveau de performance inattendu, comparable à celui obtenu précédemment sur les mêmes données avec une approche uniquement basée sur le contenu.*

ABSTRACT. *While online communities have become increasingly important over the years, the moderation of user-generated content is still performed mostly manually. Automating this task is an important step in reducing the financial cost associated with moderation, but the majority of automated approaches strictly based on message content are highly vulnerable to intentional obfuscation. In this paper, we discuss methods for extracting conversational networks based on raw multi-participant chat logs, and we study the contribution of graph features to a classification system that aims to determine if a given message is abusive. The conversational graph-based system yields unexpectedly high performance, with results comparable to those previously obtained with a content-based approach.*

MOTS-CLÉS : *Catégorisation de texte, Détection d'abus, Communautés en ligne, Modération*

KEYWORDS: *Text categorisation, Abuse detection, Online communities, Moderation*

1. Introduction

L'accès à Internet est aujourd'hui extrêmement répandu, ce qui permet à de nombreux utilisateurs du monde entier de se rencontrer et d'échanger ensemble, formant

ainsi des communautés en ligne dont le nombre ne cesse de croître. De par la possibilité d'échange d'idées qu'elles offrent, ces communautés ont acquis une grande importance socio-économique. Cependant, en raison de l'anonymat qui va de pair avec ce média, ces communautés sont souvent confrontées à des comportements abusifs. Ceux-ci peuvent dégrader la qualité du service fourni, et même exposer les administrateurs à des poursuites pénales : il est donc primordial de traiter ce problème. Ceci se fait généralement par le biais de la modération, qui consiste à appliquer différentes sanctions quand un utilisateur viole les règles en vigueur dans sa communauté. Néanmoins, la réalisation manuelle de cette tâche est coûteuse, ce qui explique l'intérêt d'appliquer des méthodes automatiques. On peut distinguer deux approches : l'une semi-automatique, consistant à attirer l'attention de modérateurs humains sur certains messages susceptibles d'être abusifs ; et l'autre complètement automatique, qui détecte les messages abusifs et met en œuvre la réponse appropriée.

Dans ce travail, nous considérons cette problématique de modération automatique comme un problème de classification consistant à déterminer automatiquement si un message est abusif ou non. Pour ce faire, nous proposons une approche originale visant à explorer un ensemble de caractéristiques (*features*) calculées à partir de graphes représentant la dynamique des conversations textuelles en ligne. Nous extrayons d'abord ces réseaux conversationnels dans lesquels les nœuds correspondent à des utilisateurs et les liens à des interactions textuelles entre-eux. Nous calculons ensuite un ensemble de mesures topologiques destinées à caractériser ce réseau de différentes façons. Nous entraînons alors un classificateur sur un corpus provenant de la communauté du jeu de rôle en ligne massivement multijoueur (*MMORPG* français *SpaceOrigin* [1]). Enfin, nous menons une étude qualitative des résultats obtenus, pour analyser l'impact des features sur la performance de classification. Cet article reprend un travail précédemment présenté en anglais dans (Papegnies *et al.*, 2017a).

Dans la Section 2, nous passons rapidement en revue les travaux relatifs à la détection d'abus et à l'extraction de réseaux à partir de logs de conversations. Puis, dans la Section 3, nous décrivons notre méthode pour l'extraction de réseaux conversationnels, ainsi que les mesures topologiques utilisées pour les caractériser. Dans la Section 4, nous présentons nos données et discutons des résultats obtenus. Enfin, nous résumons les points principaux de notre travail et les pistes potentielles en Section 5.

2. Travaux connexes

Cette section passe brièvement en revue les principaux travaux relatifs à deux aspects du problème que nous traitons dans cet article. D'abord, dans la Sous-section 2.1, nous discutons des travaux visant à détecter les cas d'abus en ligne. Puis, dans la Sous-section 2.2, nous nous tournons vers les méthodes visant à extraire des graphes à partir de données représentant des conversations.

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

2.1. Détection d'abus

On peut distinguer deux types de méthodes principaux pour la détection d'abus : celles qui utilisent le contenu des messages échangés, et celles qui se concentrent sur leur contexte. Il faut ajouter à cela les approches hybrides combinant les deux. Pour ce qui est du premier type, on peut notamment citer (Spertus, 1997), qui constitue une première tentative de classification de messages hostiles, grâce à des règles déclenchées par des marqueurs linguistiques prédéfinis. Notons cependant que la présence seule d'un de ces mots considéré comme hostile ne permet pas d'affirmer de manière certaine que le message est abusif. Pour pallier ce problème, des travaux se sont concentrés sur la prise en compte du contexte d'apparition des mots, comme dans (Chen *et al.*, 2012), où les auteurs ont adopté une approche à base de n-grammes de mots. Dans (Chavan, Shylaja, 2015), les auteurs montrent que l'occurrence de pronoms, qui est souvent négligée en classification de texte, est importante pour ce problème-ci, et l'utilisent en conjonction avec des features à base de n-grammes ainsi que de skip-grammes. Les approches traitant du contenu constituent généralement une bonne référence en termes de performance, et elles sont généralement peu coûteuses computationnellement. Cependant, elles présentent aussi des limites importantes : un abus peut s'étendre sur une succession de messages ; certains messages peuvent faire référence à une histoire commune aux participants, et il est facile de modifier l'orthographe d'une insulte pour échapper aux filtres lexicaux (Hosseini *et al.*, 2017).

Pour ces raisons, certains auteurs considèrent les messages situés *aux alentours* du message à classifier, et plus généralement son contexte. Dans (Yin *et al.*, 2009), des features sont ainsi calculées pour les phrases avoisinant le message à traiter. Il est aussi possible de construire des modèles d'utilisateur, comme dans (Cheng *et al.*, 2015). Ces auteurs se sont pour cela basés sur une étude comportementale des utilisateurs présentant des traits anti-sociaux en ligne. Dans (Balci, Salah, 2015), les auteurs utilisent des features décrivant les utilisateurs (genre, nombre d'amis, classement sur la plateforme, etc.) afin d'aider les modérateurs à traiter les abus. Dans notre propre travail précédent (Papegnies *et al.*, 2017b), nous avons proposé une approche mélangeant des features dérivées du contenu (sac-de-mots, scores $tf\text{-}idf$, scores de sentiment, etc.) et d'autres relatives au contexte, prenant la forme de modèles de langage des utilisateurs.

2.2. Extraction de réseaux

Très peu d'auteurs se sont intéressés à l'extraction de réseaux conversationnels, indépendamment de leur application à la détection d'abus. Ceci s'explique par le fait que la tâche est loin d'être triviale, et dépend grandement de la nature des données disponibles. Par exemple, on traitera différemment des logs de chat, qui sont peu ou prou des séquences de messages, et des messages de forums, qui ont la particularité d'avoir des structures arborescentes. Pourtant, ce type de réseau est particulièrement intéressant, car il permet de représenter la façon dont les utilisateurs interagissent, d'une façon bien différente de ce qui est fait avec les features mentionnées précédemment.

Dans (Mutton, 2004), les auteurs travaillent sur des logs de chat IRC et veulent simplement visualiser les interactions des utilisateurs. Pour extraire le réseau, ils utilisent un ensemble de règles simples basées sur la mention explicite de noms d'utilisateurs, ainsi que sur la proximité et la densité temporelles des messages. Dans (Tavassoli *et al.*, 2014), les auteurs travaillent aussi sur des logs de chat, et explorent plusieurs approches. En particulier, ils appréhendent les mentions de noms d'utilisateur de façon floue, afin de tenir compte de la variation orthographique, et définissent des règles pour détecter les cas où un utilisateur s'adresse à tout l'auditoire. Dans (Sinha, Rajasingh, 2014), les auteurs s'intéressent encore à des logs de chat, en se basant sur des mentions explicites de noms d'utilisateurs. Ils appliquent d'ailleurs une méthode relativement similaire pour traiter les fautes d'orthographes. Ils définissent en plus des classes comportementales d'utilisateurs, en se basant sur une partition de leur réseau social, obtenu par ailleurs.

3. Méthodes

Notre approche consiste à extraire un réseau conversationnel à partir de logs de chat, de calculer différentes mesures topologiques pour le caractériser, et de les utiliser comme features pour entraîner un classificateur. La partie classification étant standard, nous nous concentrons ici sur le réseau : la Sous-section 3.1 décrit notre méthode d'extraction et la Sous-section 3.2 se concentre sur les mesures topologiques sélectionnées.

3.1. Extraction du réseau

Nous extrayons des réseaux représentant des conversations entre plusieurs utilisateurs, via un canal de chat. Ils prennent, dans un premier temps, la forme de graphes pondérés non-orientés, dont les nœuds représentent les utilisateurs et les liens leurs interactions. Le poids d'un lien est une estimation de l'intensité de la communication ayant lieu entre les deux utilisateurs concernés. Un réseau distinct doit être extrait pour chaque *message ciblé*, i.e. chaque message que l'on veut classifier, puisque le but de l'opération est de produire des features utilisables par le classificateur.

La première étape consiste à déterminer quels messages utiliser pour extraire le réseau. Pour cela, nous définissons une *période de contexte* : un intervalle temporel centré sur l'occurrence du *message ciblé*. Dans un premier temps, nous utilisons arbitrairement une taille de 200 messages. Les graphes extraits de cette période de contexte contiennent uniquement les nœuds représentant des utilisateurs qui ont posté au moins un message sur le canal de chat traité durant l'intervalle de temps correspondant.

La deuxième étape porte sur l'ajout des liens appropriés au réseau pour l'instant vide, et sur le calcul de leurs poids. Nous utilisons pour cela une méthode à base de fenêtre glissante. Ceci est justifié par deux propriétés de l'interface graphique du chat traité : 1) quand un utilisateur rejoint un canal, le serveur lui envoie seulement les 20 messages postés ; et 2) l'historique des messages affichés ne dépasse pas 20

lignes. Nous utilisons, là encore arbitrairement, une fenêtre de 10 messages. Nous appliquons un processus itératif consistant à faire glisser la fenêtre sur toute la période de contexte, par pas de 1 message. Nous appelons *message courant* le *dernier* message de la fenêtre considérée à un instant donné. Notre hypothèse est que ce message est destiné aux auteurs des autres messages présents dans la fenêtre à ce moment-là. De plus, nous supposons que plus un auteur a posté récemment, et plus il est probable que le message lui soit adressé. Ces hypothèses se justifient par une autre propriété de l'interface graphique : par défaut, les utilisateurs ne savent pas qui est présent sur un canal à un moment donné. En particulier, ils n'ont pas connaissance de qui rejoint ou quitte le canal.

Le calcul des poids est directement basé sur ces hypothèses. D'abord, nous listons les auteurs des messages présents dans la fenêtre courante, les ordonnons par dernier message posté, et écartons l'auteur du *message courant* (plusieurs de ses messages pourraient être présents dans cette fenêtre). Nous obtenons alors une *liste de voisins*. Cependant, l'interface graphique permet de mentionner *explicitement* des utilisateurs dans un message, ce qui doit être pris en compte. Pour cela, nous déplaçons en tête de la liste les utilisateurs référencés explicitement dans le *message courant*. Si un tel utilisateur n'était même pas présent dans la fenêtre, il est simplement inséré en tête de liste. Chaque utilisateur de cette liste reçoit un score calculé au moyen d'une fonction décroissante de sa position dans la liste et de la longueur de la liste. Nous pouvons alors mettre à jour le graphe : nous créons un lien entre chaque utilisateur de la liste et l'auteur du *message courant*, doté d'un poids correspondant au score associé à l'utilisateur. Si ce lien existe déjà, son poids est simplement augmenté de ce score.

Notre décision de créer ou mettre à jour des liens attachés à tous les utilisateurs présents dans la fenêtre, même en cas de référencement direct, est basée sur plusieurs considérations. Premièrement, la mention explicite ne signifie pas forcément que l'utilisateur référencé participe à la conversation ou que le message lui est adressé. Par exemple, son nom pourrait avoir été simplement cité en tant qu'objet de la phrase. Deuxièmement, on peut rencontrer plusieurs références directes dans un seul message. Troisièmement, dans le cadre d'échanges en ligne, une personne mentionnée explicitement n'est pas forcément la seule destinataire du message. Par exemple, dans une discussion politique, une question adressée explicitement à un utilisateur peut avoir pour objectif secondaire que celui-ci expose ses idées à l'auditoire.

Une fois que le processus itératif a été appliqué à l'intégralité de la période de contexte, nous obtenons ce que nous appelons le réseau *Full*. Afin de réaliser des tests supplémentaires, nous extrayons également 2 réseaux plus réduits, basés sur le même contexte : les réseaux *Before* et *After*, construits uniquement sur les 100 messages précédant ou suivant le *message ciblé* (ainsi que ce message lui-même). La Figure 1 montre les trois réseaux ainsi obtenus pour un cas dans lequel le *message ciblé* est abusif.

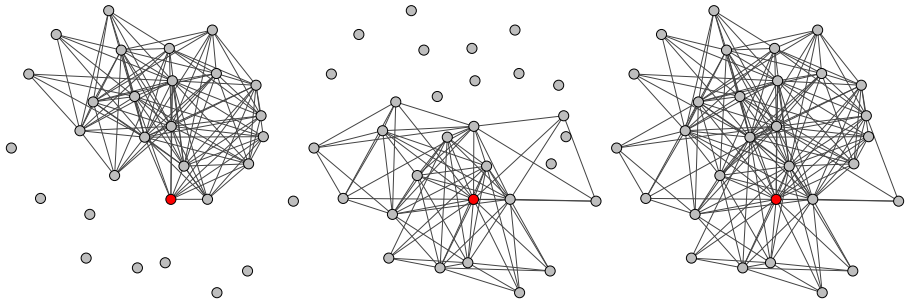

FIGURE *1. Exemples de réseaux conversationnels obtenus pour un contexte donné. De gauche à droite : Before, After et Full. L'utilisateur ayant produit le message abusif est représenté en rouge.*

3.2. Caractéristiques sélectionnées

Les caractéristiques (features) que nous utilisons pour la classification sont toutes des mesures topologiques permettant de caractériser le réseau conversationnel de différentes façons. Nous adoptons ici une approche exploratoire et considérons les mesures les plus répandues dans la littérature. Nous distinguons les mesures *locales* qui décrivent un nœud individuellement, et les *globales* qui traitent du graphe entier. Chacune est calculée pour les 3 types de réseaux décrits dans la section précédente (*Before*, *After*, *Full*).

3.2.1. Mesures topologiques locales

Chacune de ces mesures est calculée pour le nœud correspondant à l'auteur du *message ciblé*.

La *Centralité de Degré* est une version normalisée du degré standard. La *Centralité Spectrale* (Bonacich, 1987) (Eigenvector) tient son nom du fait qu'elle est basée sur le spectre de la matrice d'adjacence du graphe. Elle peut être considérée comme une généralisation du degré dans laquelle, au lieu de seulement compter le nombre de voisins d'un nœud, on tient aussi compte de leur propre centralité.

La *Centralité PageRank* (Brin, Page, 1998) est aussi spectrale, mais se base sur une variante de la matrice d'adjacence, relative à une marche aléatoire parcourant le réseau. Les *Score de Hub* et *Score d'autorité* (Kleinberg, 1999) sont deux mesures complémentaires utilisant un autre type de marche aléatoire.

La *Centralité d'Intermédiarité* (Freeman, 1978) (Betweenness) repose sur le nombre de plus courts chemins passant par le nœud considéré. Dans les réseaux de communication, elle est parfois interprétée comme le niveau de contrôle exercé par le nœud considéré sur la transmission de l'information dans le système. La *Centralité de Proximité* (Freeman, 1978) (Closeness) est l'inverse de la somme des distances géodésiques entre le nœud considéré et les autres nœuds du graphe. On considère généra-

lement qu'elle quantifie l'efficacité du nœud à faire circuler un message sur le graphe, ainsi que son indépendance vis-à-vis des autres nœuds en termes de communication. L'*Excentricité* (Harary, 1969) est aussi basée sur la distance, mais à la différence des autres mesures sélectionnées, elle évalue combien le nœud considéré est périphérique, en considérant la distance au nœud qui en est le plus éloigné.

Enfin, le *Score de Coreness* (Seidman, 1983) est basé sur la notion de k-core, qui est un sous-graphe induit maximal dont tous les nœuds ont un degré d'au moins k. La coreness d'un nœud est la valeur k du k-core de degré maximal auquel il appartient.

3.2.2. Mesures topologiques globales

Nous utilisons tout d'abord des statistiques très classiques décrivant la taille du graphe : le *Nombre de Nœuds* et le *Nombre de Liens*. Il faut ajouter à cela la *Densité*, qui est la proportion de liens existant dans le graphe considéré, par rapport à un graphe complet de même taille.

Nous utilisons aussi deux mesures à base de distance. La première est le *Diamètre*, qui correspond à la plus grande distance trouvée dans le graphe (i.e. la longueur du plus long plus court chemin). La seconde est la *Distance Moyenne*, i.e. la moyenne des distances entre toutes les paires de nœuds constituant le graphe.

Nous calculons le *Nombre de Cliques* dans le réseau, une clique étant un sous-graphe complet. L'*Assortativité de Degré* (Newman, 2002) est aussi potentiellement intéressante. Il s'agit de la corrélation entre les séries constituées des degrés des nœuds connectés, et elle mesure la dépendance statistique entre le degré d'une paire de nœuds et la présence d'un lien entre eux. Enfin, nous calculons également la valeur moyenne, pour le graphe, de chacune des 10 mesures locales présentées précédemment.

4. Validation expérimentale

Nous présentons maintenant brièvement notre corpus et notre dispositif expérimental (Sous-section 4.1), avant de décrire et de discuter nos résultats de classification (Sous-section 4.2).

4.1. Données

Nous avons accès à une base de données comportant 4 029 343 messages échangés par les utilisateurs d'un jeu vidéo en ligne massivement multijoueur. Parmi ces messages, 779 ont été signalés comme abusifs par au moins un utilisateur, ce signalement ayant ensuite été confirmé par un modérateur du jeu. Ils constituent la classe *Abus*. Chacun de ces messages est apparu dans un canal de communication différent. La classe *OK* est obtenue en échantillonnant aléatoirement 2 000 messages parmi ceux n'ayant pas été signalés comme étant abusifs. Ce même corpus a été utilisé dans notre travail précédent, qui exploitait exclusivement le contenu textuel pour réaliser la classification (Papegnies *et al.*, 2017b).

En raison de la taille relativement restreinte de notre corpus, notre protocole expérimental repose sur une validation croisée en 10 parties. Nous utilisons Python-iGraph (Csardi, Nepusz, 2006) pour extraire le réseau et calculer les features de chaque message. Pour ce qui est du classificateur, nous appliquons l'implémentation de SVM incluse dans Sklearn sous le nom SVC (C-Support Vector Classification) (Pedregosa *et al.*, 2011).

4.2. Résultats

Le Tableau 1 présente les résultats obtenus pour une référence aléatoire, pour la méthode que nous avons proposée précédemment et qui exploite à la fois le contenu textuel et le contexte des messages (Papegnies *et al.*, 2017b), et pour l'approche originale présentée ici, qui repose sur les seules features extraites du graphe conversationnel. La référence aléatoire utilise le même classificateur et la même architecture que cette dernière, excepté le fait que le calcul de feature est remplacé par un tirage au sort de valeurs dans $[0, 1]$. Notre méthode précédente exploite des features morphologiques, linguistiques et comportementales, telles que : longueur du message, nombre de mots, taux de compression, sac-de-mots & score tf–idf, et probabilité d'émission de n-grammes. Notons que le corpus a été ré-échantillonné, ce qui explique la légère différence entre les performances obtenues dans ce travail et celles présentées dans notre article précédent (Papegnies *et al.*, 2017b).

Par rapport à notre approche précédente, les features à base de graphe permettent d'améliorer la performance selon les 3 mesures considérées (Précision, Rappel, F-mesure). La performance globale est bien plus élevée que ce à quoi nous nous attendions pour une méthode qui *ignore complètement les contenus* échangés par les utilisateurs. Nous supposons que ceci s'explique principalement par le fait que les deux tiers des features utilisées incluent de l'information décrivant la partie de la conversation qui se déroule *après* le message classifié, alors que c'était le cas pour seulement 2 features (sur 67) dans notre approche à base de contenu/contexte. Indépendamment de cela, le fait que ces deux méthodes atteignent des niveaux de performance relativement élevés est un résultat très prometteur : étant donné que les deux classificateurs sont basés sur des features très différentes, il est raisonnable de supposer que leur combinaison devrait amener une amélioration supplémentaire de la performance globale.

Tableau 1. Résultats de classification (en %) pour les 3 classificateurs : référence aléatoire, méthode précédemment proposée, et méthode proposée dans cet article. Toutes les mesures sont calculées pour la classe Abus.

Classificateur	**Précision**	**Rappel**	**F-mesure**
Référence aléatoire	29,3	52,6	37,6
Features contenu/contexte	70,3	74,3	72,2
Features graphe	76,8	77,2	77,0

Comme notre classificateur est un SVM, nous pouvons utiliser l'implémentation du Platt Scaling proposée dans Sklearn pour faire varier le seuil de décision et ainsi

calibrer le système pour favoriser la précision ou le rappel. Le graphique de gauche de la Figure 2 montre les courbes Précision-Rappel pour les 10 instances de classificateurs entraînés au cours de notre expérience. On peut observer que le fait de diminuer marginalement le seuil de décision entraîne une meilleure couverture de la classe *Abus* sans pour autant perdre trop de précision. La méthode semble donc plus adaptée à une modération semi-automatique, consistant à attirer l'attention de modérateurs humains sur des messages à risque (plutôt qu'une modération automatique sans intervention humaine).

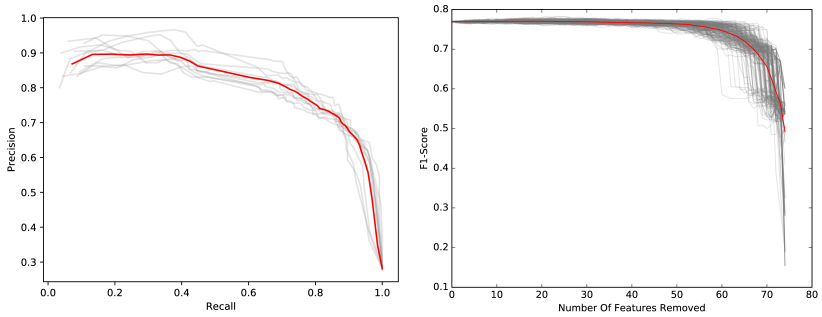

FIGURE 2. *À gauche : courbes Précision-Rappel pour les* 10 *instances de classificateurs SVM. À droite : dégradation de la performance par suppression séquentielle des features (200 répétitions). Dans les deux graphiques, la courbe en trait rouge épais représente la moyenne.*

Afin d'estimer l'importance de nos features par rapport à cette tâche de classification, nous avons utilisé le méta-estimateur *ExtraTreesClassifier* inclus dans Sklearn. Cet outil stochastique permet d'estimer pour chaque feature un score indiquant sa contribution au processus de classification. Sur la base de cette information, nous avons ré-entraîné le SVM itérativement, sur un nombre décroissant de features : en partant de l'ensemble complet des features, nous supprimons une feature à chaque itération, par ordre de score croissant. Pour chaque classificateur entraîné, nous avons ensuite calculé sa performance, qui est représentée dans le graphique de droite de la Figure 2. Celle-ci montre donc comment la performance évolue lorsqu'on considère de moins en moins de features, en commençant par les moins discriminantes. Le Tableau 2 présente les 10 features les plus discriminantes d'après nos résultats : en utilisant seulement ces features-ci, il est possible d'obtenir une F-mesure de $75,8\%$.

Globalement, ces features-là sont assez hétérogènes, topologiquement parlant, dans le sens où elles correspondent à des manières très différentes de caractériser la structure d'un graphe. La *Centralité de Degré*, le *Nombre de Liens*, et la *Densité* sont basées sur une vue microscopique du graphe : nœuds et liens sont considérés individuellement, ou seulement en termes de voisinage direct. Au contraire, la *Centralité d'Intermédiarité*, le *Score de Hub* et l'*Excentricité* sont macroscopiques car elles tirent parti de chemins susceptibles de traverser le graphe tout entier. Finalement, le *Score de Coreness* est mésoscopique, dans le sens où il est basé sur une vue intermédiaire et

Tableau 2. Features les plus discriminantes pour l'approche à base de graphe.

Graphe	**Feature**	F**-mesure *avant* suppr.**
Full	Centralité d'Intermédiarité moyenne	75,8
Before	Score de Coreness moyen	75,4
After	Nombre de Liens	74,5
After	Densité	73,1
Full	Score de Hub	72,9
After	Centralité de Degree	67,7
Before	Nombre de Liens	67,2
Full	Excentricité moyenne	58,4
Before	Centralité Spectrale moyenne	56,6
Full	Excentricité	35,0

considère des sous-graphes. Ces observations sont consistantes avec l'hypothèse que les features redondantes ne devraient pas apparaître parmi les plus discriminantes.

À première vue, il peut être surprenant de trouver à la fois le *Nombre de Liens* et la *Densité* parmi ces features : la dernière étant une version normalisée de la première, on pourrait supposer qu'elles sont redondantes. Cependant, cette normalisation est basée sur le nombre de nœuds dans le graphe. Dans le cas présent, cela signifie donc simplement que le nombre de liens de nos réseaux n'augmente pas en fonction du carré de leur nombre de nœuds. Au contraire, certaines features présentes dans la table forment (avec d'autres features absentes de la table) des groupes de features très corrélées, qui pourraient quasiment être considérées comme interchangeables. Par exemple, la corrélation entre la *Centralité Spectrale moyenne* et le *Score de Hub moyen* du grapeh *Before* s'élève à 0.73.

Les 3 types de graphes considérés (*Before*, *After*, *Full*) sont tous représentés dans la table, ce qui signifie qu'ils encodent des informations différentes et apportent tous une certaine contribution au processus de classification. De plus, il apparaît que certaines features équivalentes ou corrélées apparaissent ensemble pour plusieurs types de graphe. C'est par exemple le cas du *Nombre de Liens* (*Before* vs. *After*), et du *Score de Hub* et de la *Centralité Spectrale* (*Full* vs. *Before*). Nous supposons que ceci reflète le fait qu'un abus modifie significativement le déroulement de la conversation, et donc la structure du graphe, du moins telle qu'elle est décrite par ces features. En d'autres termes, ces features révèlent les changements importants qui surviennent dans la dynamique conversationnelle. Quand une feature apparaît uniquement pour le graphe *Before* or *After*, nous pouvons conclure qu'elle ne permet que de caractériser l'état de la conversation avant ou après l'abus.

Il est intéressant de constater que la feature *Excentricité* apparaît dans le tableau à la fois dans sa version individuelle et moyenne. Un examen approfondi montre que leurs valeurs sont plus faibles pour les graphes de la classe *Abus*. Ceci signifie que la distance maximale entre l'auteur du *message ciblé* et le reste du graphe décroît en cas d'abus. Plus concrètement, cet utilisateur devient moins périphérique (ou plus

central), et on peut dire la même chose des autres utilisateurs (en moyenne). Ceci est consistant avec la conception que l'on peut avoir de l'impact d'un abus sur une discussion : l'utilisateur au comportement abusif aura tendance à se placer au premier rang des débats, et à attirer les foudres des autres participants. Ceci pourrait expliquer pourquoi ces features sont, de loin, les plus discriminantes.

5. Conclusion

Dans cet article, nous avons présenté une approche pour classifier des messages abusifs en se basant uniquement sur des features extraites de graphes conversationnels. La méthode, bien que simple, produit des résultats raisonnablement bons, et même meilleurs que nos travaux précédents, qui exploitaient uniquement le contenu et le contexte des messages. Elle a en outre l'avantage d'être indépendante du langage traité. Cependant, il est important de relever deux limites. D'une part, le coût computationnel est plus élevé qu'avec notre approche précédente. D'autre part, la méthode ne fonctionne qu'après la publication d'un certain nombre de réponses au message abusif, ce qui empêche tout utilisation en temps réel. Notre prochaine étape dans ce projet sera d'évaluer différentes variantes de notre méthode d'extraction du réseau : attribution des poids, taille de la période de contexte et de la fenêtre glissante, etc. Nous comptons ensuite combiner les deux types de features (contenu et graphe) dans un même classificateur.

Remerciements

Ce travail a été financé par la région Provence Alpes Côte d'Azur et la société Nectar de Code.

Bibliographie

Balci K., Salah A. A. (2015). Automatic analysis and identification of verbal aggression and abusive behaviors for online social games. *Computers in Human Behavior*, vol. 53, p. 517-526.

Bonacich P. F. (1987). Power and centrality: A family of measures. *American Journal of Sociology*, vol. 92, p. 1170-1182.

Brin S., Page L. E. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, vol. 30, p. 107-117.

Chavan V. S., Shylaja S. S. (2015). Machine learning approach for detection of cyber-aggressive comments by peers on social media network. In *IEEE ICACCI*, p. 2354-2358.

Chen Y., Zhou Y., Zhu S., Xu H. (2012). Detecting offensive language in social media to protect adolescent online safety. In *PASSAT/SocialCom*, p. 71-80.

Cheng J., Danescu-Niculescu-Mizil C., Leskovec J. (2015). Antisocial behavior in online discussion communities. *arXiv:1504.00680 [cs.SI]*.

Csardi G., Nepusz T. (2006). The igraph software package for complex network research. *InterJournal Complex Systems*, vol. 1695, n° 5, p. 1-9.

Freeman L. C. (1978). Centrality in social networks I: Conceptual clarification. *Social Networks*, vol. 1, n° 3, p. 215-239.

Harary F. (1969). *Graph theory*. Addison-Wesley.

Hosseini H., Kannan S., Zhang B., Poovendran R. (2017). Deceiving google's perspective API built for detecting toxic comments. *arXiv:1702.08138 [cs.LG]*.

Kleinberg J. (1999). Authoritative sources in a hyperlinked environment. *Journal of the Association for Computing Machinery*, vol. 46, n° 5, p. 604-632.

Mutton P. (2004). Inferring and visualizing social networks on internet relay chat. In *8th international conference on information visualisation*, p. 35-43.

Newman M. E. J. (2002). Assortative mixing in networks. *Physical Review Letters*, vol. 89, n° 20, p. 208701.

Papegnies E., Labatut V., Dufour R., Linarès G. (2017a). Graph-based features for automatic online abuse detection. In *5th international conference on statistical language and speech processing*.

Papegnies E., Labatut V., Dufour R., Linarès G. (2017b). Impact of content features for automatic online abuse detection. In *International conference on computational linguistics and intelligent text processing*.

Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O. *et al*. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, vol. 12, p. 2825-2830.

Seidman S. B. (1983). Network structure and minimum degree. *Social Networks*, vol. 5, n° 3, p. 269-287.

Sinha T., Rajasingh I. (2014). Investigating substructures in goal oriented online communities: Case study of Ubuntu IRC. In *IEEE international advance computing conference*, p. 916-922.

Spertus E. (1997). Smokey: Automatic recognition of hostile messages. In *14th national conference on artificial intelligence and 9th conference on innovative applications of artificial intelligence*, p. 1058-1065.

Tavassoli S., Moessner M., Zweig K. A. (2014). Constructing social networks from semi-structured chat-log data. In *IEEE/ACM international conference on advances in social networks analysis and mining*, p. 146-149.

Yin D., Xue Z., Hong L., Davison B. D., Kontostathis A., Edwards L. (2009). Detection of harassment on Web 2.0. In *WWW workshop: Content analysis in the Web 2.0*.