Toxicité et Sentiment : Comment l'étude des sentiments peut aider la détection de toxicité

Mémoire

Eloi Brassard-Gourdeau

Maîtrise en informatique - avec mémoire
Maître ès sciences (M. Sc.)

Québec, Canada

Toxicité et Sentiment
Comment l'étude des sentiments peut aider la détection de toxicité

Mémoire

Éloi Brassard-Gourdeau

Sous la direction de:

Richard Khoury, directeur de recherche

Résumé

La détection automatique de contenu toxique en ligne est un sujet très important aujourd'hui. Les modérateurs ne peuvent filtrer manuellement tous les messages et les utilisateurs trouvent constamment de nouvelles façons de contourner les filtres automatiques. Dans ce mémoire, j'explore l'impact que peut avoir la détection de sentiment pour améliorer trois points importants de la détection automatique de toxicité : détecter le contenu toxique de façon plus exacte ; rendre les filtres plus difficiles à déjouer et prédire les conversations les plus à risque. Les deux premiers points sont étudiés dans un premier article, où l'intuition principale est qu'il est plus difficile pour un utilisateur malveillant de dissimuler le sentiment d'un message que certains mots-clés à risque. Pour tester cette hypothèse, un outil de détection de sentiment est construit, puis il est utilisé pour mesurer la corrélation entre sentiment et toxicité. Par la suite, les résultats de cet outil sont utilisés comme caractéristiques pour entraîner un modèle de détection de toxicité, et le modèle est testé à la fois dans un contexte classique et un contexte où on simule des altérations aux messages faites par un utilisateur tentant de déjouer un filtre de toxicité. La conclusion de ces tests est que les informations de sentiment aident à la détection de toxicité, particulièrement dans un contexte où les messages sont modifiés. Le troisième point est le sujet d'un second article, qui a comme objectif de valider si les sentiments des premiers messages d'une conversation permettent de prédire si elle va dérailler. Le même outil de détection de sentiments est utilisé, en combinaison avec d'autres caractéristiques trouvées dans de précédents travaux dans le domaine. La conclusion est que les sentiments permettent d'améliorer cette tâche également.

Abstract

Automatic toxicity detection of online content is a major research field nowadays. Moderators cannot filter manually all the messages that are posted everyday and users constantly find new ways to circumvent classic filters. In this master's thesis, I explore the benefits of sentiment detection for three majors challenges of automatic toxicity detection: standard toxicity detection, making filters harder to circumvent, and predicting conversations at high risk of becoming toxic. The two first challenges are studied in the first article. Our main intuition is that it is harder for a malicious user to hide the toxic sentiment of their message than to change a few toxic keywords. To test this hypothesis, a sentiment detection tool is built and used to measure the correlation between sentiment and toxicity. Next, the sentiment is used as features to train a toxicity detection model, and the model is tested in both a classic and a subversive context. The conclusion of those tests is that sentiment information helps toxicity detection, especially when using subversion. The third challenge is the subject of our second paper. The objective of that paper is to validate if the sentiments of the first messages of a conversation can help predict if it will derail into toxicity. The same sentiment detection tool is used, in addition to other features developed in previous related works. Our results show that sentiment does help improve that task as well.

Table des matières

Liste des tableaux

Liste des figures

Remerciements

Tout d'abord merci à mon directeur, Richard Khoury, pour son soutien constant en recherche et en rédaction, ses idées et sa constante disponibilité. Merci à toute l'équipe TwoHat, notamment Chris Priebe, Laurence Brockman et Anne-Marie Thérien-Daniel, ainsi qu'à tous les stagiaires avec qui j'ai eu le plaisir de travailler, Andre Schorlemmer, Talia Sanchez Viera, Jonathan Gingras, Marc-André Larochelle, Zeineb Trabelsi et Charles Poitras. Merci aux évaluateurs de mon mémoire, Luc Lamontagne et François Laviolette.

Merci à ma conjointe et ma famille, qui on pu m'aider et me conseiller même s'ils ne comprenaient pas du tout ce que je faisais.

Avant-propos

Ce mémoire contient deux articles rédigés par moi, Éloi Brassard-Gourdeau, avec le soutien de mon directeur, Richard Khoury. Je suis l'auteur principal des deux articles et j'ai effectué toutes les expérimentations et analyses présentées. Mon coauteur m'a conseillé lors de la recherche et de la rédaction.

Le premier article, *Subversive Toxicity Detection using Sentiment Information*, a été soumis au *Third Abusive Language Worhshop* le 3 mai 2019, accepté le 24 mai 2019 et publié le 31 juillet 2019. Le second article, *Using Sentiment Information for Predictive Moderation in Online Conversations*, a été soumis au *Thirty-Fourth AAAI Conference on Artificial Intelligence* le 2 septembre 2019.

Introduction

Les communautés en ligne sont de plus en plus importantes de nos jours. Chaque jour, plus de 32 millions de Canadiens et trois milliards de personnes à travers le monde sont en contact avec d'autres individus sur internet. Que ce soit sur un site de nouvelles, un réseau social, un forum de jeux vidéo ou autre, il est possible d'avoir un très grand nombre d'interactions différentes très rapidement et facilement. Cependant, cette grande ouverture n'a pas que des impacts positifs. La distance qu'il y a entre les usagers et le nombre d'interactions par jours ont permis la création d'une nouvelle façon de communiquer où les conventions sont complètement différentes.

Beaucoup d'utilisateurs se croient tout permis dans leurs interactions en ligne, car ils ne voient pas les conséquences de leurs paroles, il est très facile de quitter rapidement une conversation ou un échange et internet confère un certain anonymat. On retrouve dans les communications en ligne une modération très peu efficace, et énormément de contenu toxique.

Ce que nous voulons dire par contenu toxique, c'est une des grandes variétés de types de commentaires qui sont néfastes pour la communauté en général. Certains types de toxicité sont très graves, comme par exemple la prédation sexuelle, la radicalisation et l'incitation au suicide. Par contre, la majorité des comportements dits toxiques s'apparentent au "trolling". Ce type de comportement peut avoir une impact très léger, mais il peut parfois dégénérer en cyber-intimidation, et complètement détruire des communautés en ligne.

Pour tenter de combattre ces comportements toxiques et garder leur communauté saine et populaire, plusieurs sites ont des modérateurs. Étant donné le très grand débit de contenu généré sur les communautés en ligne (il y a par exemple plus de 3 millions de commentaires par jour sur Reddit [1]), les modérateurs ne peuvent pas espérer traiter tous ces messages et supprimer ceux qui sont dangereux. Il est donc nécessaire d'automatiser une bonne partie du processus, pour que les modérateurs n'aient seulement qu'une très petite partie des messages à gérer.

Dans cette optique, il est intéressant de trouver des outils permettant de classer les messages comme toxiques ou non-toxiques. Le but de ce projet de recherche est donc de se servir des techniques d'apprentissage automatique modernes pour aider la détection de messages dangereux dans les communautés en ligne.

1. https://foundationinc.co/lab/reddit-statistics/

Ce sujet de recherche est devenu très populaire et plusieurs chercheurs se sont déjà penchés sur le problème durant les dernières années. Les travaux marquants seront couverts dans les sections 1.4 et 2.4. Cependant, plusieurs difficultés sont soulignées dans l'état de l'art et parmi celles-ci, deux nous intéressent particulièrement : la difficulté de gérer les utilisateurs tentant de déjouer le système (la subversion du système) et la capacité à prédire quelles conversations sont à risque (la modération prédictive). Bien qu'à première vue, ces deux défis semblent très différent l'un de l'autre, certaines techniques peuvent être appliquées aux deux avec succès.

Une de ces techniques est la détection de sentiment, et il est en effet possible de se servir des connaissances préétablies dans ce domaine pour aider la détection de toxicité. Dans le premier cas, l'intuition principale est que même s'il est assez facile de dissimuler les principaux mots toxiques d'un message, il est plus difficile de cacher le sentiment toxique. Ceci est le sujet de l'article présenté au Chapitre 1. Dans cet article, il est tout d'abord question de construire un bon outil de détection de sentiment à partir de l'état de l'art, puis de vérifier la corrélation entre sentiment et toxicité pour confirmer que le premier peut être indicateur du deuxième, pour finalement s'en servir pour aider la détection de toxicité.

Dans le second cas, l'objectif est d'observer le sentiment des premiers messages d'une conversation, avec le même outil développé dans le premier article, puis de ce servir de cette information pour prédire si la suite de la conversation est à haut risque de devenir toxique. Ce point est le sujet du second article, présenté au Chapitre 2. Dans ce second article, on se concentre tout d'abord sur la validation que l'on peut bien se servir des informations de sentiment pour prédire si une conversation est à risque, puis sur l'analyse des résultats en profondeur pour bien comprendre leur impact et la meilleure façon de s'en servir.

Chapitre 1

Subversive Toxicity Detection using Sentiment Information

1.1 Résumé

La toxicité des interactions est devenu un problème majeur pour plusieurs communautés en lignes. Les modérateurs tentent de limiter ce problème en implémentant des filtres de plus en plus raffinés, mais les utilisateurs malveillants trouvent constamment de nouvelles façons de les déjouer. Notre hypothèse est que bien qu'il soit facile de modifier et dissimuler des expression très toxiques, de cacher le sentiment du message est plus difficile. Dans cet article, nous explorons différents aspects de la détection de sentiments et leur corrélation avec la toxicité, et nous utilisons nos résultats pour implémenter un outil de détection de toxicité. Nous testons ensuite comment l'ajout d'informations de sentiments aide à détecter la toxicité dans trois jeux de données réels, et nous simulons les modifications d'un utilisateur subversif tentant de déjouer le système. Nos résultats montrent que les sentiments ont un impact positif sur la détection de toxicité.

1.2 Abstract

The presence of toxic content has become a major problem for many online communities. Moderators try to limit this problem by implementing more and more refined comment filters, but toxic users are constantly finding new ways to circumvent them. Our hypothesis is that while modifying toxic content and keywords to fool filters can be easy, hiding sentiment is harder. In this paper, we explore various aspects of sentiment detection and their correlation to toxicity, and use our results to implement a toxicity detection tool. We then test how adding the sentiment information helps detect toxicity in three different real-world datasets, and incorporate subversion to these datasets to simulate a user trying to circumvent the system. Our results show sentiment information has a positive impact on toxicity detection.

1.3 Introduction

Online communities abound today, forming on social networks, on webforums, within videogames, and even in the comments sections of articles and videos. While this increased international contact and exchange of ideas has been a net positive, it has also been matched with an increase in the spread of high-risk and toxic content, a category which includes cyberbullying, racism, sexual predation, and other negative behaviors that are not tolerated in society. The two main strategies used by online communities to moderate themselves and stop the spread of toxic comments are automated filtering and human surveillance. However, given the sheer number of messages sent online every day, human moderation simply cannot keep up, and either leads to a severe slowdown of the conversation (if messages are pre-moderated before posting) or allows toxic messages to be seen and shared thousands of times before they are deleted (if they are post-moderated after being posted and reported). In addition, human moderation cannot scale up easily to the number of messages to monitor ; for example, Facebook has a team of 20,000 human moderators, which is both massive compared to the total of 25,000 other employees in the company, and minuscule compared to the fact its automated algorithms flagged messages that would require 180,000 human moderators to review [1]. Keyword detection, on the other hand, is instantaneous, scales up to the number of messages, and prevents toxic messages from being posted at all, but it can only stop messages that use one of a small set of denied words, and are thus fairly easy to circumvent by introducing minor misspellings (i.e. writing "kl urself" instead of "kill yourself"). In (Hosseini et al., 2017), the authors show how minor changes can elude even complex systems. These attempts to bypass the toxicity detection system are called subverting the system, and toxic users doing it are referred to as subversive users.

In this paper, we consider an alternative strategy for toxic message filtering. Our intuition is that, while high-risk keywords can easily be disguised, the negative emotional tone of the message cannot. Consequently, we will study the correlation between sentiment and toxicity and its usefulness for toxic message detection both in subversive and non-subversive contexts. It is important to note that toxicity is a very abstract term that can have different definitions depending on context, and each dataset described in Section 1.6 has its own. They all gravitate around negative messages such as insults, bullying, vulgarity and hate speech, therefore these types of toxic behavior are the ones we focus on, as opposed to other types such as fraud or grooming that would use more positive messages.

The rest of this paper is structured as follows. After a review of the relevant literature in the next section, we will consider the problem of sentiment detection in online messages in Section 1.5. We will study the measure of toxicity and its correlation to message sentiment in Section 1.6. Finally, we will draw some concluding remarks in Section 1.7.

1. http://fortune.com/2018/03/22/human-moderators-facebook-youtube-twitter/

1.4 Related Work

Given the limitations of human and keyword-based toxicity detection systems mentioned previously, several authors have studied alternative means of detecting toxicity. In one of the earliest works on the detection of hate speech, the authors of (Warner and Hirschberg, 2012) used n-grams enhanced by part-of-speech information as features to train an SVM classifier to accurately pick out anti-semitic online messages. Following a similar idea, the authors of (Nobata et al., 2016) conducted a study of the usefulness of various linguistic features to train a machine learning algorithm to pick out hate speech. They found that the most useful single feature was character n-grams, followed closely by word n-grams. However, it was a combination of all their features (n-grams, features of language, features of syntax, and word embedding vectors) that achieved the highest performance. The authors of (Alorainy et al., 2018) studied hate speech through the detection of othering language. They built a custom lexicon of pronouns and semantic relationships in order to capture the linguistic differences when describing the in-group and out-group in messages, and trained a word embedding model on that data.

Hate speech is not the only form of toxicity that has been studied. In (Reynolds et al., 2011), the authors studied cyberbullying. They developed a list of 300 "bad" words sorted in five levels of severity. Next, they used the number and density of "bad" words found in each online message as the features to train a set of machine learning systems. The authors of (Ebrahimi, 2016) also used words as features in two systems, this time to detect sexual predators. One used the TFxIDF values of the words of the text to train a single-class SVM classifier, and the other used a bag-of-words vector of the text as input to a deep neural network. The authors found that the latter system offered the better performance in their experiments.

Recently, deep learning has become very popular for NLP applications, and pre-trained word embeddings have been shown to be very effective in most text-based neural network applications. In (Agrawal and Awekar, 2018), four different deep learning models were implemented and shown to outperform benchmark techniques for cyberbullying detection on three different datasets. In (Chatzakou et al., 2017), a deep neural network taking a word embedding vector as input was used to detect cyberbullying on Twitter.

It thus appears from the related literature that authors have tried a variety of alternative features to automatically detect toxic messages without relying strictly on keyword detection. However, sentiment has rarely been considered. It was one of the inputs of the deep neural network of (Chatzakou et al., 2017), but the paper never discussed its importance or analyzed its impact. The authors of (Hee et al., 2018) conducted the first study of cyberbullying in Dutch, and considered several features, including a subjectivity keyword lexicon. They found its inclusion helped improve results, but that a more sophisticated source of information than simple keyword detection was required. And the study of (Dani et al., 2017) used the sentiment of messages, as measured by the SentiStrength online system, as one of several features to detect cyberbullying messages. However, an in-depth analysis of how sentiment

can benefit toxicity detection has not been done in any of these papers, and a study of the use of sentiment in a subversive context has never been done.

1.5 Sentiment Detection

1.5.1 Lexicons

Sentiment detection, or the task of determining whether a document has a positive or negative tone, has been frequently studied in the literature. It is usually done by using a sentiment lexicon that either classifies certain words as positive or negative, or quantifies their level of positivity or negativity. We decided to consider six such lexicons :

— **SentiWordNet** [2] is a widely-used resource for sentiment mining. It is based on WordNet, and assigns three scores to each synset, namely positivity, negativity, and objectivity, with the constraint that the sum of all three must be 1. Using this lexicon requires a bit of preprocessing for us, since the same word can occur in multiple different synsets with different meanings and therefore different scores. Since picking out the intended meaning and synset of a polysemous word found in a message is beyond our scope, we instead chose to merge the different meanings and compute a weighted average of the scores of the word. The weights are the ranks of the synsets, which correspond to the popularity of that meaning of the word in documents. The average score equation is :

$$score = \frac{\sum^{k} \frac{score}{rank}}{\sum^{k} \frac{1}{rank}} \tag{1.1}$$

where k is the number of times the word occurs with the same part of speech. We compute the average positivity and negativity scores, but not the objectivity scores, since they are not useful for our purpose and since they are simply the complement of the other two. This allows us to extract 155,287 individual words from the lexicon, with a positivity and negativity score between 0 and 1 for each. We should note that SentiWordNet differentiates a word based on part-of-speech, and we maintain this distinction in our work.

— **Afinn** [3] is a lexicon of 3,382 words that are rated between -5 (maximum negativity) and 5 (maximum positivity). To match SentiWordNet, we split this score into positivity and negativity scores between 0 and 1. For example, a word with a -3 score was changed to have a positive score of 0 and a negative score of 0.6.

— **Bing Liu** [4] compiled lists of 6,789 positive or negative words. Given no other information, we assigned each word in the positive list a positivity score of 1 and a negativity score of 0, and vice-versa for the negative-list words.

2. http://sentiwordnet.isti.cnr.it/
3. https://github.com/fnielsen/afinn
4. https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

Word	SentiWordNet	Afinn	Bing Liu	General Inquirer	Subjectivity Clues	NRC
terrorize	[0.125, 0.250]	-3	negative	negative	strong negative	negative
helpless	[0.000, 0.750]	-2	negative	negative	weak negative	negative
joke	[0.375, 0.000]	2	negative	positive	strong positive	negative
merry	[0.250, 0.250]	3	positive	positive	strong positive	positive
splendid	[1.000, 0.000]	3	positive	positive	strong positive	positive

TABLE 1.1 – Sentiment of words per lexicon

— **General Inquirer**[5] is a historically-popular lexicon of 14,480 words, though only 4,206 of them are tagged as positive or negative. As for the Bing Liu lexicon, we assigned binary positive and negative scores to each word that was tagged as positive or negative.

— **Subjectivity Clues**[6] extends the sentiment tags of the General Inquirer to 8,222 words using a dictionary and thesaurus. It also adds a binary strength level (strong or weak) to the polarity information. We merged polarity and strength as a measure of 0.5 and 1 for weak or strong positivity or negativity.

— **NRC**[7] has a list of 14,182 words that are marked as associated (1) or not associated (0) with 8 emotions (anger, fear, anticipation, trust, surprise, sadness, joy, disgust) and two sentiments (negative and positive). We transform this association into binary positive and negative scores in the same way we did for Bing Liu and General Inquirer.

All six of these lexicons have limitations, which stem from their limited vocabulary and the ambiguity of the problem. Indeed, despite being thousands of words each and covering the same subject and purpose, our six lexicons have only 394 words in common, indicating that each is individually very incomplete compared to the others. And we can easily find inconsistencies between the ratings of words, both internally within each lexicon and externally when we compare the same words between lexicons. Table 1.1 illustrates some of these inconsistencies : for instance, the word "helpless" is very negative in SentiWordNet but less so in Afinn and Subjectivity Clues, while the word "terrorize" is more strongly negative in the latter two resources but less negative (and even a bit positive) in SentiWordNet. Likewise, the word "joke" is strongly positive, weakly positive, or even negative, depending on the lexicon used, and the word "merry" is more positive than "joke" according to every lexicon except SentiWordnet, which rates it equally positive and negative. By contrast the word "splendid" has the same positivity values as "merry" in all lexicons except SentiWordnet, where it has the highest possible positivity score.

In a longer document, such as the customer reviews these lexicons are typically used on (Ohana et al., 2012; Tumsare et al., 2014; Agarwal et al., 2015), these problems are minor : the abundance and variety of vocabulary in the text will insure that the correct sentiment emerges overall despite the noise these issues cause. This is not true for the short messages of online conversations, and it has

5. http://www.wjh.harvard.edu/~inquirer/
6. http://mpqa.cs.pitt.edu/lexicons/
7. https://nrc.canada.ca/en/

forced some authors who study the sentiments of microblogs to resort to creating or customizing their own lexicons (Nielsen, 2011). This, incidentally, is also why we could not simply use an existing sentiment classifier. We will instead opt to combine these lexicons into a more useful resource.

1.5.2 Message Preprocessing

The first preprocessing step is to detect the presence and scope of negations in a message. Negations have an important impact; the word "good" may be labeled positive in all our lexicons, but its actual meaning will differ in the sentences "this movie is good" and "this movie is not good". We thus created a list of negation keywords by combining together the lists of the negex algorithm [8] and of (Carrillo de Albornoz et al., 2012), filtering out some irrelevant words from these lists, and adding some that were missing from the lists but are found online.

Next, we need to determine the scope of the negation, which means figuring out how many words in the message are affected by it. This is the challenge of, for example, realizing that the negation affects the word "interesting" in "this movie is not good or interesting" but not in "this movie is not good but interesting". We considered two algorithms to detect the scope of negations. The first is to simply assume the negation affects a fixed window of five words [9] after the keyword (Councill et al., 2010), while the second discovers the syntactic dependencies in the sentence in order to determine precisely which words are affected (Dadvar et al., 2011).

We tested both algorithms on the SFU review corpus of negation and speculation [10]. As can be seen in Table 1.2, the dependency algorithm gave generally better results, and managed to find the exact scope of the negation in over 43% of sentences. However, that algorithm also has a larger standard deviation in its scope, meaning that when it fails to find the correct scope, it can be off by quite a lot, while the fixed window is naturally bounded in its errors. Moreover, the increased precision of the dependencies algorithm comes at a high processing cost, requiring almost 30 times longer to analyze a message as the fixed window algorithm. Given that online communities frequently deal with thousands of new messages every second, efficiency is a major consideration, and we opted for the simple fixed window algorithm for that reason.

The second preprocessing step is to detect sentiment-carrying idioms in the messages. For example, while the words "give" and "up" can both be neutral or positive, the idiom "give up" has a clear negative sentiment. Several of these idioms can be found in our lexicons, especially SentiWordNet (slightly over 60,000). We detect them in our messages and mark them so that our algorithm will handle them as single words going forward.

Finally, we use the NLTK wordpunkt_tokenizer to split messages into words, and the Stanford fasterEnglishPOSTagger to get the part-of-speech of each word. Since our lexicons contain only four

8. https://github.com/mongoose54/negex/tree/master/negex.python
9. The average window size in our test dataset was 5.36 words, so we rounded to the closest integer.
10. https://www.researchgate.net/publication/256766329_SFU_Review_Corpus_Negation_Speculation

	Fixed window	Dependencies
Accuracy	71.75%	82.88%
Recall	95.48%	90.00%
Precision	69.65%	78.37%
Exact match	9.03%	43.34%
Std	3.90 words	5.54 words
ms/sentence	2.4	68

TABLE 1.2 – Comparison between fixed window and syntactic dependencies negation detection algorithms

parts-of-speech (noun, verb, adverb, and adjective) and Stanford's tagger has more than 30 possible tags, we manually mapped each tag to one of the four parts-of-speech (for example, "verb, past participle" maps to "verb").

1.5.3 Message Sentiment

Once every word has a positivity and a negativity score, we can use them to determine the sentiment of an entire message. We do this by computing separately the sum of positive scores and of negative scores of words in the message, and subtracting the negative total from the positive total. In this way, a score over 0 means a positive message, and a score under 0 means a negative message. We consider two alternatives at this point : one in which we sum the sentiment value of all words in the message, and one where we only sum the sentiment value of the top-three [11] words with the highest scores for each polarity. We label these "All words" and "Top words" in our results. The impact of this difference is felt when we consider a message with a few words with a strong polarity and a lot of words with a weak opposite polarity ; in the "Top words" scheme these weak words will be ignored and the strong polarity words will dictate the polarity of the message, while in the "All words" scheme the many weak words can sum together to outweigh the few strong words and change the polarity of the message.

We optionally take negations into account in our sentiment computation. When a word occurs in the window of a negation, we flip its positivity and negativity scores. In other words, instead of adding its positivity score to the positivity total of the message, we added its negativity score, and the other way round for the negativity total. For example, in SentiWordNet, the word "good" has a positive value of 0.25 and negative value of 0. However, if the word "not" is present just before, "good" will have a positive value of 0 and negative value of 0.25. Experiments where this is done are labeled "Negativity" in our results.

Finally, we optionally incorporate word weights based on their frequency in our datasets. When applied, the score of each word is multiplied by a frequency modifier, which we adapted from (Ohana

11. We considered the top-two, three, four, and five words, but early empirical tests on SentiWordNet indicated that top-three was the best option.

et al., 2012) :

$$frequency\_modifier = 1 - \sqrt{\frac{n}{n_{max}}} \tag{1.2}$$

where n is the number of times the word appears in a dataset, and n_{max} is the number of times the most frequent word appears in that dataset. Experiments using this frequency modifier are labeled “Frequency" in our results.

1.5.4 Experimental Results

Our experiments have four main objectives : (1) to determine whether the “All words" or the “Top words" strategy is preferable ; (2) to determine whether the inclusion of “Negation" and "Frequency" modifiers is useful ; (3) to determine which of the six lexicons is most accurate ; and (4) to determine whether a weighted combination of the six lexicons can outperform any one lexicon.

To conduct our experiments, we used the corpus of annotated news comments available from the Yahoo Webscope program [12]. The comments in this dataset are annotated by up to three professional, trained editors to label various attributes, including type, sentiment and tone. Using these three attributes, we split the dataset into two categories, sarcastic and non-sarcastic, and then again into five categories, clear negative, slight negative, neutral, slight positive, and clear positive. Finally, we kept only the non-sarcastic comments where all annotators agreed to reduce noise. This gives us a test corpus of 2,465 comments.

To evaluate our results, we compute the sentiment score of each comment in our test corpus using our various methods, and we then compute the average sentiment score of comments in each of the five sentiment categories. For ease of presentation, we give a simplified set of results in Table 1.3, with only the average score of the two negative and the two positive labels combined, along with the overlap of the two distributions. The overlap is obtained by taking two normal distributions with the the means and standard deviations of the positive and the negative sets, and calculating the area in common under both curves. It gives us a measure of the ambiguous region where comments may be positive or negative. A good sentiment classifier will thus have very distant positive and negative scores and a very low overlap.

These results show that there are important differences between the lexicons. Three of the six are rather poor at picking out negative sentiments, namely Subjectivity Clues (where negative messages are on average detected as more positive than the positive messages), General Inquirer, and NRC. This bias for positivity is an issue for a study on toxicity, which we expect to be expressed using negative sentiments. The other three lexicons give a good difference between positive and negative messages. For these three lexicons, we find that using *All words* increases the gap between positive and negative scores but greatly increases the standard deviation of each sentiment class, meaning the sentiment of the messages becomes ambiguous. On the other hand, using *Top words* reduces the overlap between the distributions and thus gives a better separation of positive and negative sentiments. And while

12. Dataset L32 : `https://webscope.sandbox.yahoo.com/catalog.php?datatype=l`

Experiment	SWN	Afinn	Bing Liu	Gen. Inq.	Subj. Clues	NRC
All words	[-0.22, 0.31] 0.81	[-0.43, 0.45] 0.71	[-1.17, 0.69] 0.67	[0.03, 1.44] 0.73	[2.31, 1.97] 0.76	[-0.15, 1.00] 0.77
All/Neg	[-0.34, 0.17] 0.79	[-0.44, 0.39] 0.69	[-1.08, 0.61] 0.70	[-0.27, 0.99] 0.77	[1.66, 1.52] 0.83	[-0.62, 0.75] 0.75
All/Freq	[-0.21, 0.29] 0.80	[-0.42, 0.40] 0.71	[-1.17, 0.58] 0.68	[-0.09, 1.23] 0.76	[1.98, 1.70] 0.82	[-0.19, 0.90] 0.79
All/Neg/Frq	[-0.29, 0.18] 0.78	[-0.42, 0.35] 0.69	[-1.06, 0.52] 0.71	[-0.33, 0.85] 0.79	[1.45, 1.34] 0.86	[-0.56, 0.69] 0.77
Top words	[-0.23, 0.11] 0.75	[-0.23, 0.31] 0.68	[-0.54, 0.54] 0.67	[-0.03, 0.59] 0.80	[1.18, 1.17] 0.99	[-0.14, 0.54] 0.77
Top/Neg	[-0.24, 0.10] 0.74	[-0.24, 0.29] 0.67	[-0.50, 0.53] 0.67	[-0.12, 0.57] 0.77	[0.86, 0.71] 0.94	[-0.28, 0.49] 0.73
Top/Freq	[-0.16, 0.15] 0.74	[-0.23, 0.28] 0.67	[-0.56, 0.47] 0.67	[-0.07, 0.52] 0.79	[1.00, 1.01] 0.99	[-0.15, 0.50] 0.77
Top/Neg/Frq	[-0.17, 0.14] 0.73	[-0.23, 0.26] 0.67	[-0.51, 0.48] 0.66	[-0.14, 0.49] 0.77	[0.61, 0.76] 0.93	[-0.26, 0.45] 0.74

TABLE 1.3 – Average sentiment scores of negative and positive (respectively) labeled messages, and their overlap.

Experiment	Majority vote	Maximum wins	Average scores
Top words	[-0.36, 0.34] 0.67	[-0.60, 0.52] 0.67	[-0.32, 0.32] 0.64
Top + Negation	[-0.35, 0.34] 0.66	[-0.59, 0.51] 0.66	[-0.31, 0.30] 0.63
Top + Frequency	[-0.34, 0.32] 0.66	[-0.58, 0.48] 0.67	[-0.31, 0.30] 0.63
Top + Neg. + Freq.	[-0.32, 0.30] 0.65	[-0.55, 0.50] 0.65	[-0.29, 0.29] 0.63

TABLE 1.4 – Sentiment scores using combinations of lexicons.

adding frequency information or negations does not cause a major change in the results, it does give a small reduction in overlap.

To study combinations of lexicons, we decided to limit our scope to SentiWordNet, Afinn, and Bing Liu, the three lexicons that could accurately pick out negative sentiments, and on the *Top words* strategy. We consider three common strategies to combine the results of independent classifiers : majority voting, picking the one classifier with the maximum score (which is assumed to be the one with the highest confidence in its classification), and taking the average of the scores of all three classifiers. For the average, we tried using a weighted average of the lexicons and performed a grid search to find the optimal combination. However, the best results were obtained when the three lexicons were taken equally. For the majority vote, we likewise take the average score of the two or three classifiers in the majority sentiment.

Table 1.4 presents the results we obtain with all three strategies. It can be seen that combining the three classifiers outperforms taking any one classifier alone, in the sense that it creates a wider gap between the positive and negative messages and a smaller overlap. It can also be seen that the addition of negation and frequency information gives a very small improvement in the results in all three cases. Comparing the three strategies, it can be seen that the maximum strategy gives the biggest gap in between positive and negative distribution, which was to be expected since the highest positive or negative sentiment is selected each time while it gets averaged out in the other two classifiers. However, the average score strategy creates a significantly smaller standard deviation of sentiment scores and a lower overlap between the distributions of positive and negative messages. For that reason, we find the average score to be the best of the three combination strategies.

In all cases, we find that most misclassified messages in our system are due to the lack of insults in the

vocabulary. For example, none of the lexicons include colorful insults like “nut job" and “fruitcake", so messages where they appear cannot be recognized as negative. Likewise, some words, such as the word “gay", are often used as insults online, but have positive meanings in formal English ; this actually leads to labeling insult messages as positive. This issue stems from the fact that these lexicons were designed for sentiment analysis in longer and more traditional documents, such as customer reviews and editorials. One will seldom, if ever, find insults (especially politically-incorrect ones such as the previous examples) in these documents.

1.6 Toxicity Detection

With a good and adatpable sentiment detection tool, we can now focus on the main contribution of this paper : to study how sentiment can be used to detect toxicity in subversive online comments. To do this, we will use three new test corpora :

— The **Reddit** [13] dataset is composed of over 880,000 comments taken from a wide range of subreddits and annotated a few years ago by the *Community Sift* tool developed by *Two Hat Security* [14]. This toxicity detection tool, which was used in previous research on toxicity as well (Mohan et al., 2017), uses over 1 million n-gram rules in order to normalize then categorize each message into one of eight risk levels for a wide array of different categories, 0 to 3 being super-safe to questionable, 4 being unknown and 5 to 7 being mild to severe. In our case, we consider the scores assigned to each message in five categories, namely bullying, fighting, sexting, vulgarity, and racism.

— The **Wikipedia Talk Labels** [15] dataset consists of over 100,000 comments taken from discussions on English Wikipedia’s talk pages. Each comment was manually annotated by around ten Crowdflower workers as toxic or not toxic. We use the ratio of toxic marks as a toxicity score. For example, if a message is marked toxic by 7 out of 10 workers, it will have a 0.7 toxicity score.

— The **Kaggle toxicity competition** [16] dataset is also taken from discussions on English Wikipedia talk pages. There are approximatively 160,000 comments, which were manually annotated with six binary labels : toxic, severe_toxic, obscene, threat, insult, and identity_hate. This allows us to rate comments on a seven-level toxicity scale, from 0/6 labels marked to 6/6 labels marked.

1.6.1 Correlation

Our first experiment consists in computing the sentiment of each message in each of our three test corpora, and verifying how they correlate with the different toxicity scores of each of the corpora. Following the results we found in Section 1.5, we used the best three lexicons (SentiWordNet, Afinn,

13. https://bigquery.cloud.google.com/table/fh-bigquery:reddit_comments.2007
14. https://www.twohat.com/community-sift/
15. https://figshare.com/articles/Wikipedia_Talk_Labels_Toxicity/4563973
16. https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

Sentiment	Reddit	Wikipedia	Kaggle
Standard	-0.2410	-0.3839	-0.3188
Negation	-0.2021	-0.3488	-0.2906
Frequency	-0.2481	-0.3954	-0.3269
Neg + Freq	-0.2056	-0.3608	-0.3003

TABLE 1.5 – Correlation between sentiment and toxicity.

and Bing Liu), combined them by taking the average score, and used our four algorithm variations. The results are presented in Table 1.5. It can be seen that there is a clear negative correlation between toxicity and sentiment in the messages, as expected. Our results also show that using words only or including frequency information makes the relationship clearer, while adding negations muddies it. These results are consistent over all three test corpora, despite being from different sources and labeled using different techniques. The lower score on the Reddit dataset may simply be due to the fact it was labeled automatically by a system that flags potentially dangerous content and not by human editors, so its labels may be noisier. For example, mentioning sexual body parts will be labeled as toxicity level 5 even if they are used in a positive message, because they carry more potential risk.

1.6.2 Subversive Toxicity Detection

Our second experiment consists in studying the benefits of taking sentiments into account when trying to determine whether a comment is toxic or not. The toxicity detector we implemented in this experiment is a deep neural network inspired by the most successful systems in the Kaggle toxicity competition we used as a dataset. It uses a bi-GRU layer with kernel size of 40. The final state is sent into a single linear classifier. To avoid overfitting, two 50% dropout layers are added, one before and one after the bi-GRU layer.

The network takes as input a message split into words and into individual characters. The words are represented by the 300d fastText pre-trained word embeddings [17], and characters are represented by a one-hot character encoding but restricted to the set of 60 most common characters in the messages to avoid the inclusion of noise. The character embeddings enrich the word embeddings and allow the system to extract more information from the messages, especially in the presence of misspellings (Shen et al., 2017). Finally, we used our “top + frequency" sentiment algorithm with the best three lexicons (SentiWordNet, Afinn, and Bing Liu) to determine the sentiment of each message. We input that information into the neural network as three sentiment values, corresponding to each of the three lexicons used, for each of the frequent words retained for the message. Words that are not among the selected frequent words or that are not found in a lexicon receive a sentiment input value of 0. Likewise, experiments that do not make use of sentiment information have inputs of 0 for all words. These input values are then concatenated together into a vector of 363 values, corresponding to the 300 dimensions of fastText, the 60 one-hot character vector, and the 3 sentiment lexicons. We can see

17. https://github.com/facebookresearch/fastText

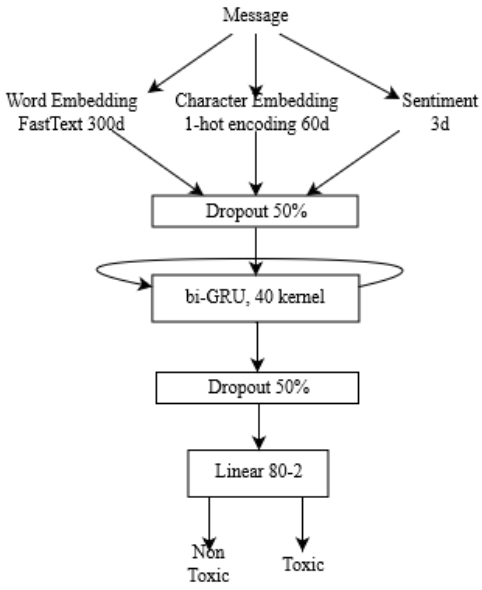

FIGURE 1.1 – Model architecture

diagram of the model in Figure 1.1.

The output of our network is a binary “toxic or non-toxic" judgment for the message. In the Kaggle dataset, this corresponds to whether the “toxic" label is active or not. In the Reddit dataset, it is the set of messages evaluated at levels 5, 6 or 7 by *Community Sift* in any of the topics mentioned earlier. And in the Wikipedia dataset, it is any message marked as toxic by 5 workers or more. We chose this binary approach to allow the network to learn to recognize toxicity, as opposed to types of toxic messages on Kaggle, keyword severity on Reddit, or a particular worker’s opinions on Wikipedia. However, this simplification created a balance problem : the Reddit dataset is composed of 12% toxic messages and 88% non-toxic messages, the Wikipedia dataset is composed of 18% toxic messages, and the Kaggle dataset of 10% toxic messages. To create balanced datasets for training, we kept all toxic messages and undersampled randomly the set of non-toxic messages to be equal to the number of toxic messages. This type of undersampling is commonplace in order to avoid the many training issues that stem from heavily imbalanced datasets.

Our experiment consists in comparing the toxicity detection accuracy of our network when excluding or including sentiment information and in the presence of subversion. Indeed, as mentioned in Sections 1.3 and 1.4, it is trivial for a subversive user to mask toxic keywords to bypass toxicity filters. In order to simulate this behavior and taking ideas from (Hosseini et al., 2017), we created a substitution list that replaces popular toxic keywords with harmless versions. For example, the word “kill" is replaced by “kilt", and “bitch" by “beach". Our list contains 191 words, and its use adds noise to 82% of the toxic Kaggle messages, 65% of the Wikipedia messages, and 71% of the Reddit messages. These substitutions are only done at testing time, and not taken into account in training, to simulate the fact that users can create never-before-seen modifications.

Dataset	Standard			Sentiment			*p-value*
	Accuracy	Precision	Recall	Accuracy	Precision	Recall	
Kaggle	93.2%	93.1%	93.0%	93.7%	92.1%	95.2%	0.0188
Subv. Kaggle	77.2%	93.3%	58.8%	80.1%	94.1%	65.6%	<0.0001
Wiki	88.1%	87.4%	89.1%	88.5%	87.9%	89.4%	0.0173
Subv. Wiki	81.4%	86.1%	75.5%	82.0%	86.3%	75.9%	0.0165
Reddit	94.0%	98.2%	89.6%	94.1%	98.3%	89.7%	0.4159
Subv. Reddit	87.1%	98.0%	75.9%	88.0%	98.2%	77.5%	0.0098

TABLE 1.6 – Accuracy, precision and recall on regular and subversive datasets, with and without sentiment, along with the *t-test p-value* when comparing accuracy result distribution

We trained and tested our neural network with and without sentiment information, with and without subversion, and with each corpus three times to mitigate the randomness in training. In every experiment, we used a random 70% of messages in the corpus as training data, another 20% as validation data, and the final 10% as testing data. The average results of the three tests are given in Table 1.6. We performed a *t-test* on the accuracy result distribution to determine if the difference between the results with and without sentiment information is statistically significant, and the *p-value* is also included in Table 1.6. As a reminder, the *t-test* compares the two distributions to see if they are different from each other, and assigns a *p-value* to this result. As a general rule, a *p-value* below 0.05 indicates that the *t-test* found a statistically significant difference between the two distributions.

It can be seen that sentiment information helps improve toxicity detection in a statistically-significant manner in all cases but one. The improvement is smaller when the text is clean (without subversion). In those experiments, the accuracy improvement is of 0.5% or less. However, the introduction of subversion leads to an important drop in the accuracy of toxicity detection for the network that uses the text alone. Most of that loss comes from a much lower recall score, which is unsurprising considering the fact that we are modifying the most common toxic words. The inclusion of sentiment information makes it possible to mitigate that loss. With subversion, including sentiment information improves the accuracy of toxicity detection by more than 0.5% in all experiments, and as much as 3% on the Kaggle dataset, along with a decrease in *p-value* in all cases.

For example, the message “The bot sucks. No skills. Shut it down." isn’t detected as toxic after adding subversion, because the toxic word “sucks" is changed to the harmless word “socks". However, when including sentiment information, the system detects the negative tone of the message - with the “No skills. Shut it down." part being clearly negative - and increases the score sufficiently for the message to be classified as toxic. Sentiment information is also helpful even in the absence of subversion. For example, the message “You make me sick to my stomach, whoever you are and whatever your motivations might be. You have caused an odious stench which will be impossible to erase." lacks recognizable toxic features such as insults and curse words and is classified as non-toxic by the sentiment-less neural network. However, the negative sentiment of “sick", “stench", and “odious" (none of which are normally found in abusive word lists) allows the sentiment neural network to recognize the message

as toxic.

Comparing the different corpora, it can be seen that the improvement is smallest and least significant in the Reddit dataset experiment, which was to be expected since it is also the dataset in which toxicity and sentiment had the weakest correlation in Table 1.5. We can note that our toxicity detection neural network performs very well nonetheless in all cases, even with subversion and without sentiment information. This may be due to the fact that the messages in all datasets are user-generated and therefore noisy already. In addition, the character encoding of the neural network is robust to misspellings, as opposed to a keyword lookup system. The results are also very close to the top solutions of the Kaggle competition for the Kaggle dataset with a 98.1 AUC (top solutions being 98.8) while taking a lot less time to train and not using huge manual misspellings lists or data augmentation like all top solutions do.

1.7 Conclusion

In this paper, we explored the relationship between sentiment and toxicity in social network messages. We began by implementing a sentiment detection tool using different lexicons and different features such as word frequencies and negations. This tool allowed us to demonstrate that there exists a clear correlation between sentiment and toxicity. Next, we added sentiment information to a toxicity detection neural network, and demonstrated that it does improve detection accuracy. Finally, we simulated a subversive user who circumvents the toxicity filter by masking toxic keywords in their messages, and found that using sentiment information improved toxicity detection by as much as 3%. This confirms our fundamental intuition, that while it is possible for a user to mask toxic words with simple substitutions, it is a lot harder for a user to conceal the sentiment of a message.

Our work so far has focused on single-line messages and negative toxicity detection. There are however several different types of toxicity, some of which correlate to different sentiments. For instance, fraud or sexual grooming will use more positive sentiments in order to lure victims. Differentiating between these types of toxicity will strengthen the correlation to message sentiment and further improve our results. Likewise, handling entire conversations will allow us to include contextual information to the sentiment of each message, and to detect sudden changes in the sentiment of the conversation that correspond to a disruptive toxic comment.

Acknowledgment

This research was made possible by the financial, material, and technical support of Two Hat Security Research Corp., and the financial support of the Canadian research organization MITACS.

Bibliographie

Basant Agarwal, Namita Mittal, Pooja Bansal, and Sonal Garg. Sentiment analysis using common-sense and context information. *Computational intelligence and neuroscience*, 2015 :30, 2015.

Sweta Agrawal and Amit Awekar. Deep learning for detecting cyberbullying across multiple social media platforms. *CoRR*, abs/1801.06482, 2018. URL http://arxiv.org/abs/1801.06482.

Wafa Alorainy, Pete Burnap, Han Liu, and Matthew Williams. Cyber hate classification :'othering'language and paragraph embedding. *arXiv preprint arXiv :1801.07495*, 2018.

Jorge Carrillo de Albornoz, Laura Plaza, Alberto Dìaz, and Miguel Ballesteros. Ucm-i : A rule-based syntactic approach for resolving the scope of negation. In **SEM 2012 : The First Joint Conference on Lexical and Computational Semantics – Volume 1 : Proceedings of the main conference and the shared task, and Volume 2 : Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 282–287. Association for Computational Linguistics, 2012. URL http://www.aclweb.org/anthology/S12-1037.

Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Athena Vakali. Mean birds : Detecting aggression and bullying on twitter. *CoRR*, abs/1702.06877, 2017. URL http://arxiv.org/abs/1702.06877.

Isaac G Councill, Ryan McDonald, and Leonid Velikovich. What's great and what's not : learning to classify the scope of negation for improved sentiment analysis. In *Proceedings of the workshop on negation and speculation in natural language processing*, pages 51–59. Association for Computational Linguistics, 2010.

Maral Dadvar, Claudia Hauff, and Franciska de Jong. Scope of negation detection in sentiment analysis. *Dutch- Belgian Information Retrieval Workshop*, 01 2011.

Harsh Dani, Jundong Li, and Huan Liu. Sentiment informed cyberbullying detection in social media. *Machine Learning and Knowledge Discovery in Databases*, pages 52–67, 01 2017.

Mohammadreza Ebrahimi. *Automatic Identification of Online Predators in Chat Logs by Anomaly Detection and Deep Learning*. PhD thesis, Concordia University, 2016.

Cynthia Van Hee, Gilles Jacobs, Chris Emmery, Bart Desmet, Els Lefever, Ben Verhoeven, Guy De Pauw, Walter Daelemans, and Véronique Hoste. Automatic detection of cyberbullying in social media text. *CoRR*, abs/1801.05617, 2018. URL http://arxiv.org/abs/1801.05617.

Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. Deceiving google's perspective API built for detecting toxic comments. *CoRR*, abs/1702.08138, 2017. URL http://arxiv.org/abs/1702.08138.

Shruthi Mohan, Apala Guha, Michael Harris, Fred Popowich, Ashley Schuster, and Chris Priebe. The impact of toxic language on the health of reddit communities. In *Canadian Conference on Artificial Intelligence*, pages 51–56. Springer, 2017.

Finn Årup Nielsen. A new anew : Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv :1103.2903*, 2011.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153. International World Wide Web Conferences Steering Committee, 2016.

Bruno Ohana, Sarah Jane Delany, and Brendan Tierney. A case-based approach to cross domain sentiment classification. In *International Conference on Case-Based Reasoning*, pages 284–296. Springer, 2012.

Kelly Reynolds, April Kontostathis, and Lynne Edwards. Using machine learning to detect cyberbullying. In *Machine learning and applications and workshops (ICMLA), 2011 10th International Conference on*, volume 2, pages 241–244. IEEE, 2011.

Yanyao Shen, Hyokun Yun, Zachary C. Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep active learning for named entity recognition. *CoRR*, abs/1707.05928, 2017. URL http://arxiv.org/abs/1707.05928.

Pranali Tumsare, Ashish S Sambare, Sachin R Jain, and Andrada Olah. Opinion mining in natural language processing using sentiwordnet and fuzzy. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) Volume*, 3 :154–158, 2014.

William Warner and Julia Hirschberg. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26. Association for Computational Linguistics, 2012.

Chapitre 2

Using Sentiment Information for Predictive Moderation in Online Conversations

2.1 Résumé

Le défi de la détection automatique de toxicité a été le sujet d'énormément de recherches récemment, mais la majorité des travaux se sont concentrés sur la mesure de toxicité d'un message après qu'il ait été partagé. Certains auteurs ont tentés de prédire si une conversation allait dérailler et devenir toxique en utilisant des caractéristiques de ses quelques premiers messages (Zhang et al., 2018). Dans cet article, nous combinons leur approche avec nos travaux précédents sur la détection de toxicité en utilisant les informations de sentiments (Brassard-Gourdeau and Khoury, 2019), et montrons comment les sentiments exprimés dans les premiers messages d'une conversation peuvent aider à prédire une suite toxique. Nos résultats montrent que l'ajout d'informations de sentiments aide à améliorer l'exactitude de la prédiction de toxicité, et nous permettent aussi de faire des observations importantes sur la tâche générale de détection préventive.

2.2 Abstract

The challenge of automatic toxicity detection has been the subject of a lot of research recently, but the focus has been mostly on measuring toxicity in individual messages after they have been posted. Some authors have tried to predict if a conversation will derail and turn toxic using the features of the first few messages (Zhang et al., 2018). In this paper, we combine that approach with our previous work on toxicity detection using sentiment information (Brassard-Gourdeau and Khoury, 2019), and show how the sentiments expressed in the first messages of a conversation can help predict a toxic outcome. Our results show that adding sentiment features does help improve the accuracy of toxicity prediction, and also allow us to make important observations on the general task of preemptive moderation.

2.3 Introduction

Billions of messages are sent online every day, and hidden among them are millions of harmful and toxic messages, making the development of accurate and efficient content moderation systems a high priority. In recent years, many researchers have studied the challenge of detecting toxic messages. However, most of these studies have focused on single line processing. In other words, the models developed look at each message individually and decided whether it should be classified as high risk or not. While such systems can be good at detecting abusive messages once they are written (Chatzakou et al., 2017; Pavlopoulos et al., 2017; Hee et al., 2018), they cannot anticipate whether a conversation will devolve into toxic messages based on context. This ability to flag interactions for moderation before they turn bad would be hugely beneficial both for community moderators, allowing them to intervene more quickly and efficiently, and for users, preventing them from being targeted by toxic comments in the first place. This is the goal of predictive moderation, or toxicity prediction. It is however impossible to do when considering only a single message in isolation.

In (Zhang et al., 2018), the authors study the pragmatic devices used early in a conversation and their effects on whether the discussion will develop in a healthy or toxic manner. In this paper, we build upon their work by adding our sentiment information (Brassard-Gourdeau and Khoury, 2019) to their model. Our intuition is that the sentiment expressed early in a conversation can help predict more accurately if it will derail or not.

The rest of this paper is structured as follows. After a review of the relevant literature in section 1.4, we will quickly go over our sentiment detection tool in Section 2.5. In the same section we will also study how our sentiment features can be added to the system of (Zhang et al., 2018) and present the resulting prediction tool. We will conduct an in-depth analysis of our results when using the dataset of (Zhang et al., 2018) in Section 2.6. To expand on this study, we then perform a second set of experiments on another dataset in Section 2.7. Finally, we draw conclusions on predictive moderation and the use of sentiment information in Section 2.8.

2.4 Related Work

The challenge of toxic content detection in online conversations has been studied since 2012. Various topics have been covered, such as hate speech detection (Warner and Hirschberg, 2012; Nobata et al., 2016) and cyberbullying detection (Reynolds et al., 2011; Chatzakou et al., 2017; Agrawal and Awekar, 2018), and many architectures have been adapted and trained successfully to this task, including SVMs (Warner and Hirschberg, 2012; Hee et al., 2018), logistic regressions (Nobata et al., 2016), and neural networks (Agrawal and Awekar, 2018). However, even the most recent work only focuses on single-line detection, meaning determining whether a comment that has already been posted is toxic or not by itself and outside the context of the conversation where it appears.

One of the first and only studies on toxicity prediction at the conversation level is that of (Zhang

et al., 2018). The authors show that certain features in the first messages of a conversation, such as the use of first or second person pronouns and the presence of certain politeness strategies, can help predict if that conversation will remain healthy or if it will degrade and lead to toxic messages later on. Their work inspired the authors of (Karan and Šnajder, 2019), who also worked on preemptive toxicity detection. They trained and tested a SVM using TFIDF-weighted unigrams and bigrams as well as a BiLSTM using their own word embeddings. They were dissatisfied with their results, however the fact they focused only on words and didn't use more sophisticated features as in (Zhang et al., 2018) may have been the cause. Finally, the authors of (Liu et al., 2018) do hostility presence and intensity forecasting on Instagram comment threads using a variety of features, ranging from n-grams and word vectors to user activity and lexicons. The features are used to train a logistic regression model with L2 regularization. The authors conclude that there are 4 main predictors for hostility : the post author's history of receiving hostile comments, the presence of user-directed profanity in the thread, the number of distinct users posting comments in that thread, and the amount of hostility so far in a conversation. However, none of these studies examined the impact of sentiment information in predictive moderation, which will be the main focus of our paper.

2.5 Conversation Model

2.5.1 Sentiment Detection Tool

We implemented a sentiment detection system in our previous work (Brassard-Gourdeau and Khoury, 2019), in order to study whether sentiment information can help detect toxic content in a subversive setting (where users deliberately misspell toxic words to mask them from keyword filters). We found that sentiment information did correlate to toxicity, and could be used to improve the accuracy of toxic message detection systems, both in a normal and in a subversive setting.

The sentiment detection tool we implemented in that paper, which we will reuse in this one, is heavily inspired by previous works such as (Ohana et al., 2012; Nielsen, 2011; Tumsare et al., 2014), where the authors used sentiment lexicons, such as SentiWordNet or General Inquirer, to detect the sentiment of a message. Our tool combines three popular lexicons, namely SentiWordNet [1], Afinn [2] and Bing Liu [3]. We found previously that these three lexicons have different strengths and weaknesses, and thus complement each other well. SentiWordNet is the biggest lexicon and assigns a positive and negative score between 0 and 1 to each word. Afinn assigns a single score between -5 and 5 ; scores under zero meaning the words are negative. The Bing Liu lexicon has a positive and a negative word list. We combined the lexicons by splitting each into lists of positive and negative words for each of four parts-of-speech (noun, verb, adverb, and adjective), and normalizing the sentiment scores between 0 and 1.

1. http://sentiwordnet.isti.cnr.it/
2. https://github.com/fnielsen/afinn
3. https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

Our system begins by detecting sentiment-carrying idioms in the messages. For example, while the words "give" and "up" can both be neutral or positive, the idiom "give up" has a clear negative sentiment. Several of these idioms can be found in our lexicons, especially SentiWordNet (slightly over 60,000). When detected, these idioms are marked so that our algorithm will handle them as single words. Next, we use the NLTK *wordpunkt_tokenizer* to split messages into words, and the *pos_tagger* to get the part-of-speech of each word. Each word is then assigned a positive and a negative score, which is the sum of the score it has in the positive and negative lists of each of the three lexicons. A message is represented by the score of its three most positive words and its three most negative words. This gives us a total of 6 sentiment features for each message. For more details as to why the tool is built this way, please refer to (Brassard-Gourdeau and Khoury, 2019).

2.5.2 Model and features

As mentionned earlier, we are using the sentiment detection tool to build upon the work of (Zhang et al., 2018). In their paper, the authors use a set of pragmatic features which they split into two categories : 13 politeness strategies and 6 rhetorical prompts. The first category focuses on the use of politeness, such as greetings, gratitude, or the use of "please", and of impoliteness, such as direct and strong disagreement or personal attacks. The second category captures six domain-specific conversation prompts, which are six clusters of conversations discovered by an unsupervised technique trained on a different dataset that includes similar types of discussions. A new conversation's distance to each of these six clusters gives the six prompt features. More details on these features can be found in the original article.

Using these features, the authors train a logistic regression model to predict if a conversation will derail into toxicity based on its first two messages. The authors have made their code available publicly [4]. The model takes as input the 13 politeness strategies and 6 rhetoric prompts features of (Zhang et al., 2018) for each of the first two messages of the conversation. To these, we added the 6 sentiment features measured by our sentiment tool for the same two messages. We also computed another sentiment feature representing the overall tone of the first two messages. This feature is computed by taking the sum of positive word scores of the first message and subtracting the sum of negative word scores to determine if the message is overall positive or negative, doing the same for the second message, and determining if the conversation starts with two positive messages, a positive followed by a negative, a negative followed by a positive, or two negative messages. This information is encoded as a one-hot vector of length 4. In total, there are thus 38 text features from (Zhang et al., 2018) and 16 sentiment features we added, for a total of 54 conversation features.

4. https ://github.com/CornellNLP/Cornell-Conversational-Analysis-Toolkit

2.5.3 Data and Training

The dataset created for (Zhang et al., 2018), which is available publicly along with their code, is a set of user conversations taken from the edit pages of English Wikipedia. The authors used Persepctive API [5] to pre-filter the conversations and keep only the ones with potentially toxic content. They furthered filtered to keep those conversations that started in a civil way, meaning that didn't have any toxic content in the first two messages. Moreover, they required conversation pairs, one derailing and one staying civil, from each Wikipedia page. This resulted in 1,270 conversation pairs from 582 different pages with an average length of 4.6 messages.

Our model is trained using *Scikit-learn*'s *LogisticRegression* and *SelectPercentile*, with a grid search on hyperparameters C between 10^{-4} and 10^{4} and *percentile* between 10 and 100 [6]. Training was done using a 5-fold cross validation. Apart from increasing the number of folds from 3 to 5 for more consistency between runs, all the training parameters are exactly the same as the ones in (Zhang et al., 2018).

2.6 Results and Analysis

As in (Zhang et al., 2018), our experiments consist in taking a pair of conversations, looking at their first two messages, and predicting which of the two conversations will remain healthy and which one will derail into toxicity. All the results presented in the following section are the average of 10 separate runs, where we randomized the data split.

2.6.1 Sentiment Features

Our first experiment considers the predictive accuracy of sentiment information alone. In fact, the authors of (Zhang et al., 2018) did include a sentiment lexicon (Liu et al., 2005) in their research, and used it to extract two sentiment features per message. Their features were "has negative" and "has positive", each being 1 if a negative or positive word from the lexicon was present in the message and 0 otherwise. However, after testing these features, they concluded that sentiment was barely better than random chance at predicting toxicity, and they didn't include these features in their set of 38 text features.

The goal of our first experiment is thus to validate that our sentiment features are in fact predictors of toxicity. We trained and tested the model using four setups : using the original sentiment features of (Zhang et al., 2018), our sentiment word features, our tone features, and all sentiment features combined. The results are presented in 2.1.

Our results firstly confirm that the minimalist sentiment features of (Zhang et al., 2018) are nearly equivalent to a random chance guess. This is likely due to the fact that over 70% of the messages

5. https ://www.perspectiveapi.com/
6. C representing the regularization and *percentile* representing the percent of features to use.

Test	Features	Accuracy
Original sentiment	4	51.3
Our sentiment	12	55.7
Our tone	8	50.8
All features	24	55.8

TABLE 2.1 – Prediction accuracy using sentiment features.

containing a negative word also have a positive word, making it nearly impossible to discern a negative message from a positive one based on that information alone. Likewise, our tone information carries nearly no useful information. However, our more detailed word features do show an interesting predictive ability. Finally, combining all features together gives no gain compared to just using our word features ; an unsurprising result, given that the other features seem to contain no predictive information.

This shows that, when it comes to sentiment information, it is not the overall sentiment of a message that is useful, but individual words. That level of detail is missing from both the original sentiment features (which only indicated whether positive or negative sentiment exist) and our tone features (which only indicate whether positive or negative sentiment is stronger). It is however present in our sentiment word features, which indicates the sentiment of the three most positive and most negative words of each message without making a judgment on whether the message overall is positive or negative. That finer level of granularity seems to be where the predictive information is found.

From this point forward, we will drop the tone features from our model, since it is not predictive of toxicity. This will leave 12 sentiment features and a total of 50 conversation features.

2.6.2 All Features

Our next test consist in training and testing our model with and without our sentiment features. The goal is to highlight the gain in prediction accuracy that comes from including sentiment features. The results of that test are given in Table 2.2.

Test	Features	Accuracy
Text features	38	58.6
Text + sentiment	50	60.5

TABLE 2.2 – Prediction accuracy with and without sentiment features.

In all 10 runs of our test, we found that the model including sentiment features consistently performs better than the one without. Our results using text features alone are consistent with those of (Zhang et al., 2018), and adding sentiment features improves the prediction on average by 2%. This is consistent with our findings in (Brassard-Gourdeau and Khoury, 2019), where we found sentiment information improved toxicity prediction by 3%.

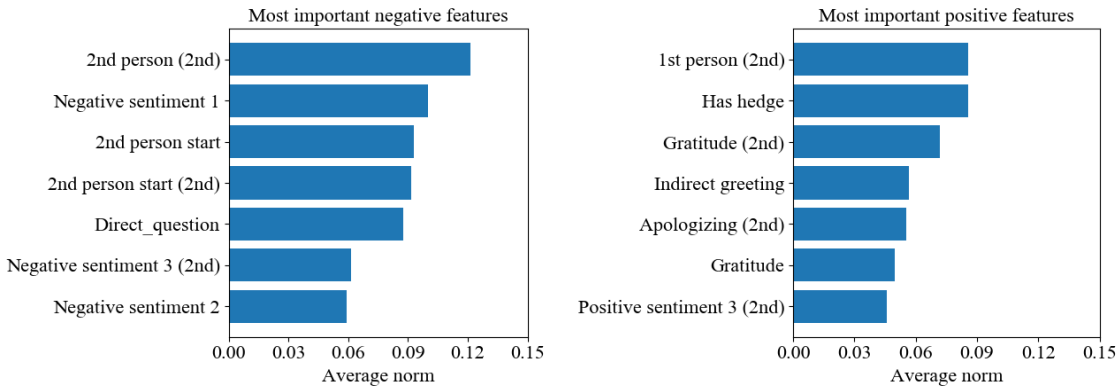

FIGURE 2.1 – Feature importance when using 3 positive sentiment features and 3 negative sentiment features. The "(2nd)" refers to the feature on the second message, while its omission refers to the first message.

2.6.3 Predictive Features

It is interesting to examine which sentiment features contribute the most information to the prediction of how a conversation will develop. To do this, we take the average norm of the coefficient score of the logistic regression for each of the 50 features over the 10 runs of our experiment. The most informative features are simply those with the highest positive or negative coefficients, while features with coefficients around 0 have no influence on the prediction.

We found that the most predictive features were consistent from run to run. They are listed in Figure 2.1, along with their average coefficients. The top text features found match those found in (Zhang et al., 2018). In addition to those, four of our sentiment features are among the 14 most predictive features found by the regression model.

For predicting conversations that will turn toxic, the strength of the first and second most negative words in the first message and of the third most negative word in the second message are all strong predictors. This indicates that strong negative words in both first messages will likely cause the conversation to degrade. Combined with the fact that second-person pronoun use in both messages are also strong toxicity predictors, this may also indicate conversations that begin with directed negative sentiments towards other participants.

On the other hand, only one of our sentiment features is among the strongest predictors of whether a conversation will remain healthy. It is the strength of the third most positive word in the second message. This is an interesting difference with the toxicity case : while strong negative words are clear predictors of upcoming toxicity, strong positive words are not predictors of health, but lower-ranked positive words are. This may indicate that abundance, not strength, of positive sentiment is what matters to predict health.

In order to verify that theory, we re-trained and re-tested our model several times using between 1

and 7 positive or negative sentiment features. The best combination we found was using 5 positive sentiment features and only 2 negative ones, and this 56-feature model offered an improvement of 1% on prediction accuracy compared to the 50-feature model of Table 2.2. The most predictive features in that test are shown in Figure 2.2. For toxicity prediction, nothing has changed, save for the fact the third negative word of the second message has disappeared (as the feature is no longer part of the model) and the second negative word of the second message is the seventh most predictive feature (it was eighth previously). For health prediction, we can see that the newly-added features of the fourth and fifth positive words of the first message are now among the top predictors, beating out the third positive word from 2.1. This confirms our earlier intuition.

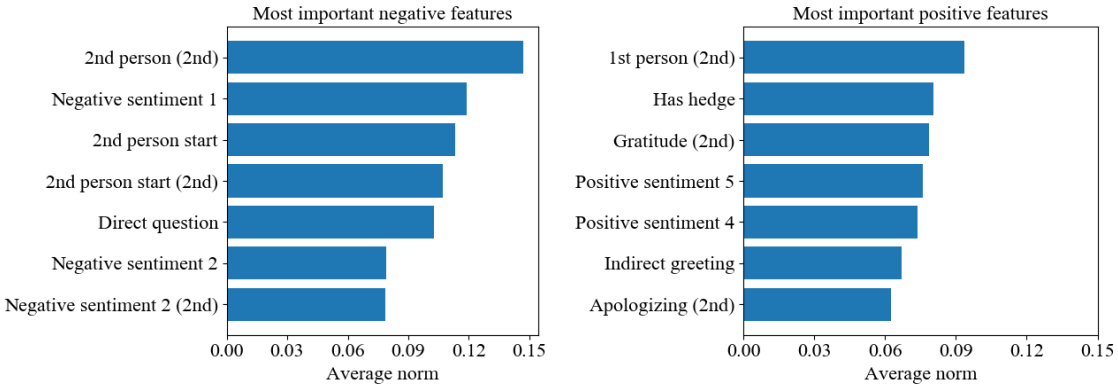

FIGURE 2.2 – Feature importance when using 5 positive sentiment features and 2 negative sentiment features.

2.6.4 Case Studies

Figure 2.3 has an example of the first two messages of a conversation that was misclassified using the text features alone, but was correctly classified by the classifier with sentiment features.

> (1) I'm sorry to say it, but I'm pretty sure this is the only option left. This discussion has been so repetitive it's unbelievable. The mediation cabal has all but ceased, and the mediation of this talk page has failed . The RFC also did not work. I can see no other way to reslve the issue other than ArbCom. What does everyone else think ?
>
> (2) It's a pretty useless process. Mostly Admins listing Hebrew as a language, or displaying Israeli symbols on their user pages will respond they have no problem with the biased edits. Worst -case they ban you for suggesting the article needed comment or some type of oversight.

FIGURE 2.3 – First two messages of a derailing conversation, with major good and bad words highlighted in green and red respectively.

The first message uses the first person and apologizes, both text features that predict a healthy conver-

sation, and no other predictive text features are present in either message. As a result, the text-based system predicts they will lead to a healthy conversation. In reality, this conversation eventually degrades into the users attacking each other with messages such as : "[username] actually blames others", "it's your problem", "you are just trying to find an excuse to take jabs at me" and eventually "[username] shut up".

When taking sentiment information into account, the picture is quite different. Both messages contain only a single strong positive sentiment word, the word "pretty" (score of 0.59). The other positive words are very weak, and the fourth and fifth positive words of the first message are "all" and "think" (scores of 0.04 and 0.02 respectively). On the other hand, the first message has two strong negative words, "failed" (score of 0.47) and "unbelievable" (score of 0.46), and the second message has three even stronger ones, "useless", "problem" and "worst" (scores of 0.67, 0.60, and 0.78 respectively). Negative features dominate these messages, and as a result our model predicts correctly that this conversation will descend into toxicity.

This example highlights one reason why the top positive words are not predictors of health : they can be used as modifiers to enhance negative words, as is the case of the word "pretty" in "pretty useless". We believe another reason the strongest positive words are not good predictors is sarcasm, which uses one or two very strongly positive words to convey a negative message. However, we found no examples of sarcasm in our dataset, so we could not confirm that hypothesis.

(1) not vandilism

(2) well sorry about replacing bands.but you dumb cunt fireworks is also a punk pop band

FIGURE 2.4 – First two messages of a derailing conversation, with major good and bad words highlighted in green and red respectively.

The sample conversation of Figure 2.3 is an example of the impact of strong negative words. The second message in particular contains an apology (positive indicator), the strong positive word "well" (score of 0.46), and uses the second person (negative indicator). However, most people will pinpoint the two negative words as the strongest indicators this conversation will degrade. In fact, if those words were removed from the message, it would become a much more civil conversation. This illustrates how one or two strongly negative words can change the tone of a message and the flow of a conversation.

2.7 Gaming Chat Moderation

To validate the generality of our results, we decided to apply our model to a completely different setting from Wikipedia talk pages : live in-game chat conversations from a popular video game [7]. This

7. The dataset was provided by Two Hat Research Corp. with permission from the gaming company. The data was pseudonymized and users had agreed to have their chat used for moderation purposes. The data can not be shared publicly

dataset consists of 26,964 different conversations of up to 50 messages, with most messages being very short, around 4 words only. This makes it very different from the Wikipedia dataset, in which conversations are on average less than 5 messages long but messages are on average 58 words long. The last message of each conversation was reported by a user, and then a decision was made by a community moderator to either take action on the reported message or ignore the report. The dataset is balanced, with 54% of messages moderated and 46% ignored.

There are several other significant differences with the Wikipedia dataset. Unlike an edit discussion which has a well-identified initial message, a gaming chat conversation begins when the chat room is created and is continuously ongoing after that, with players joining and leaving at will. The dataset's 50-message conversations are actually composed of the reported message and the previous 49 messages. Moreover, the Wikipedia dataset contains mostly two- to four-person conversations, while very often over a dozen players can chat simultaneously (together or in intertwined separate discussions) and be present in the 50-message conversation.

The purpose of this experiment is slightly different from the previous one : while we still want to determine if it is possible to predict if a conversation will become toxic (meaning in this case that it will need moderation) from earlier messages, and to measure which text and sentiment features are the strongest predictors of this, we are no longer working with conversation pairs. Consequently, instead of choosing which of two conversations is most likely to go awry, we predict for each conversation individually if it will go awry or not, which is a much harder problem. Moreover, since the first message in a conversation is not the first message of the chatroom, we are not making a prediction from the beginning of a conversation but from an arbitrary point in the middle of it. Finally, taking only the first two messages as before would represent on average 8 words, which is not enough information to make a prediction from. Consequently, we use instead the 10 messages prior to the reported comment to predict whether the unseen final message will be toxic and require moderator action or not.

We will use the same 19 text features and 7 (5 positive 2 negative) sentiment features per message as before. However, with 10 messages instead of 2, this means our model will have 260 features as input instead of 50. Moreover, we expect that message chronology will be a lot more important in a 10-message sequence than with 2 messages. Consequently, we decided to try two different models. The first one is the same logistic regression model as before. The second model is a recurrent neural network, specifically a uni-directional GRU with a kernel of 40 and a linear layer taking the final state of the GRU and producing a binary output. A recurrent neural network is a natural choice for a problem with a lot of features where chronology is important, and we used a similar model in (Brassard-Gourdeau and Khoury, 2019) for single-line toxicity detection and found it works well.

The data was randomly split 70/20/10 into training/validation/testing sets. We once again did 10 training and testing runs, using a different random split each time and 5-fold cross-validation within each run. Average results over all 10 runs are presented in Table 2.3. These results confirm that adding

due to its sensitive nature.

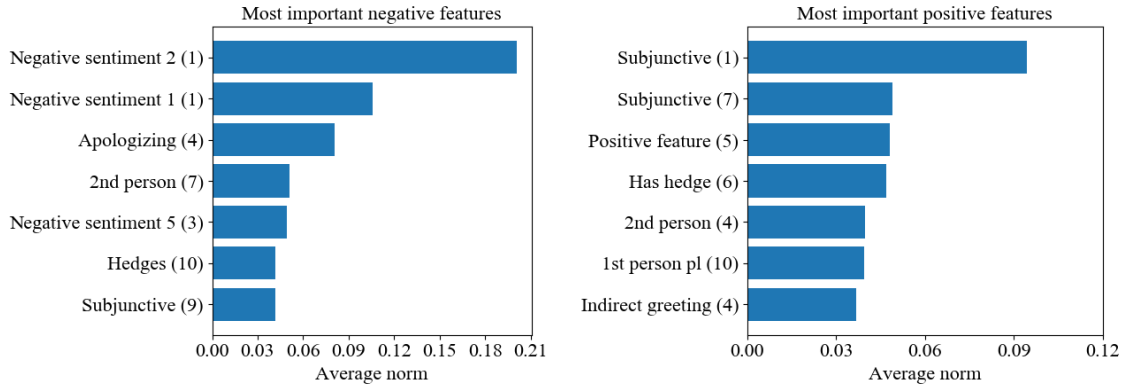

FIGURE 2.5 – Feature importance in the gaming chat dataset. The number in parenthesis refers to the message's position before the reported message.

sentiment information helps improve the prediction of toxic conversations. The gain is greater for the logistic regression model, which in fact fails to make a prediction better than random chance without sentiment information. The RNN fares better, probably because it can better handle the large number and sequential nature of the features, but it still gains 1% by including sentiment information.

Model	Features	Accuracy	F1 score
Regression	190	50.9%	0.556
Regression	260	57.4%	0.564
RNN	190	60.6%	0.686
RNN	260	61.7%	0.691

TABLE 2.3 – Results for both models using text features alone (190 features) or text and sentiment features (260 features).

As before, we used the average coefficient score of each feature over the 10 runs to rank the features by predictive importance. The top features are shown in Figure 2.5. There are some differences with the results of the Wikipedia test. Most notably, the subjunctive feature [8] is a strong predictor of healthy conversations in this experiment. Looking more closely, this feature is predictive of unmoderated conversations two-thirds of the times it appears ; however, it appears in less than 1% of chat conversations. This difference is therefore not significant in practice.

On the other hand, the coherence aspects with the previous experiment are very interesting. In both experiments, the features has hedge [9], the use of 1st person pronouns, and indirect greetings [10], are indicators of healthy conversations, while strong negative-sentiment words are indicators of upcoming toxic comments. Moreover, unlike with the 'subjunctive' feature, these features all occur in significant

8. Expressions such as "would you" and "could you".

9. 'Has hedge' refers to the presence of hedges, or mitigating words, like 'think', 'almost', 'rather', etc. This differs from the feature 'hedges', which looks for dependencies and requires the subject of the message to express this hedge.

10. The presence of words like 'hey', 'hello' or 'hi'.

numbers of the conversation. This confirms that the method is generalizable and can be applied to different types of online conversations.

Next, we considered the the question of which messages in the conversation contain the most predictive features. To this end, we considered the 26 (10%) most predictive features of health and toxicity, and grouped them per message. The results, given in Table 2.4, show that features predicting both health and toxicity can be found throughout the conversation. However, while health predictors are distributed evenly in the conversation, toxicity predictors are concentrated in the final three messages. This indicates that a healthy conversation is an ongoing process, but a few bad messages can very quickly turn the tides of the conversation and lead to toxic messages. This also indicates a limit to preemptive moderation : long-term predictions are not valid, and one must focus on clues in the latest messages. To confirm this, we ran the experiment again using only 3 messages before the reported message instead of 10. The results are almost identical to before : the logistic regression classification has an accuracy of 57.7% with sentiment and 51.7% without, while the RNN has an accuracy of 61.8% with sentiment and 60.9% without. It seems clear, then, that the previous seven messages did not contribute significantly to the classification accuracy.

Message	Health features	Toxicity features
10	3	2
9	3	2
8	3	1
7	2	3
6	1	2
5	2	1
4	4	3
3	3	6
2	3	2
1	2	4

TABLE 2.4 – Number of predictive features per message before the reported message.

2.8 Conclusion

In this paper, we studied how sentiment information can be used as a feature for the task of predictive moderation. We conducted this study using the sentiment detection tool we developed in our previous work, the conversation features and logistic classifier of (Zhang et al., 2018), and two very different online conversation datasets. The results of our experiments allow us to draw some important conclusions that can guide both future research and practical implementations of predictive moderation tools :

1. Sentiment information is indeed a predictor of toxicity. Using it improves a system's performance by between 1% and 6%, which is consistent with our previous results in (Brassard-Gourdeau and Khoury, 2019).

2. Sentiment information is found at a fine granularity, at the individual word level. Using coarser information, such as overall message sentiment, is not informative.
3. It takes a lot of weak positive words to maintain a healthy conversation, but only a few strong negative words can turn a conversation toxic.
4. The features that are predictive of health and toxicity are consistent between very different formats of conversations, and a predictive moderation system is therefore generalizable to multiple online communities.
5. A conversation turns toxic very quickly, and consequently toxicity predictors are concentrated in the few most recent messages. This puts a natural limit to the range of predictive moderation of about 3 messages.

The task of predictive moderation is still in its infancy, and there is still a lot of room for research. For example, work so far has focused on using regular conversation features as predictors. Future work could look at adding toxic text features such as insults and curse words, or even using the output of single-line toxicity detection tools as features.

Acknowledgment

This research was made possible by the financial, material, and technical support of Two Hat Security Research Corp., and the financial support of the Canadian research organization MITACS.

Bibliographie

Sweta Agrawal and Amit Awekar. Deep learning for detecting cyberbullying across multiple social media platforms. *CoRR*, abs/1801.06482, 2018. URL http://arxiv.org/abs/1801.06482.

Eloi Brassard-Gourdeau and Richard Khoury. Subversive toxicity detection using sentiment information. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 1–10, Florence, Italy, August 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W19-3501.

Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Athena Vakali. Mean birds : Detecting aggression and bullying on twitter. *CoRR*, abs/1702.06877, 2017. URL http://arxiv.org/abs/1702.06877.

Cynthia Van Hee, Gilles Jacobs, Chris Emmery, Bart Desmet, Els Lefever, Ben Verhoeven, Guy De Pauw, Walter Daelemans, and Véronique Hoste. Automatic detection of cyberbullying in social media text. *CoRR*, abs/1801.05617, 2018. URL http://arxiv.org/abs/1801.05617.

Mladen Karan and Jan Šnajder. Preemptive toxic language detection in Wikipedia comments using thread-level context. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 129–134, Florence, Italy, August 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W19-3514.

Bing Liu, Minqing Hu, and Junsheng Cheng. Opinion observer : Analyzing and comparing opinions on the web. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 342–351, New York, NY, USA, 2005. ACM. ISBN 1-59593-046-9. doi : 10.1145/1060745.1060797. URL http://doi.acm.org/10.1145/1060745.1060797.

Ping Liu, Joshua Guberman, Libby Hemphill, and Aron Culotta. Forecasting the presence and intensity of hostility on instagram using unsing linguistic and social features. In *Twelfth International AAAI Conference on Web and Social Media*, 2018.

Finn Årup Nielsen. A new anew : Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv :1103.2903*, 2011.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153. International World Wide Web Conferences Steering Committee, 2016.

Bruno Ohana, Sarah Jane Delany, and Brendan Tierney. A case-based approach to cross domain sentiment classification. In *International Conference on Case-Based Reasoning*, pages 284–296. Springer, 2012.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1125–1135, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi : 10.18653/v1/D17-1117. URL https://www.aclweb.org/anthology/D17-1117.

Kelly Reynolds, April Kontostathis, and Lynne Edwards. Using machine learning to detect cyberbullying. In *Machine learning and applications and workshops (ICMLA), 2011 10th International Conference on*, volume 2, pages 241–244. IEEE, 2011.

Pranali Tumsare, Ashish S Sambare, Sachin R Jain, and Andrada Olah. Opinion mining in natural language processing using sentiwordnet and fuzzy. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) Volume*, 3 :154–158, 2014.

William Warner and Julia Hirschberg. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26. Association for Computational Linguistics, 2012.

Justine Zhang, Jonathan P. Chang, Cristian Danescu-Niculescu-Mizil, Lucas Dixon, Yiqing Hua, Nithum Thain, and Dario Taraborelli. Conversations gone awry : Detecting early signs of conversational failure. *CoRR*, abs/1805.05345, 2018. URL http://arxiv.org/abs/1805.05345.

Conclusion

Dans ce mémoire, la relation entre les sentiments et la toxicité a été étudiée dans deux cas principaux : pour aider la détection de toxicité lorsque les utilisateurs tentent de déjouer le système, et pour la prédiction de contenu toxique dans une conversation en observant les premiers messages de celle-ci. Tout d'abord, nous avons exploré la relation entre sentiment et toxicité. Pour ce faire, nous avons implémenté un outil de détection de sentiments en utilisant plusieurs lexiques à notre disposition. Après avoir testé et comparé les différents lexiques, ainsi qu'expérimenté avec différentes stratégies d'attribution de score, nous avons combiné les lexiques et stratégies les plus performants pour créer le meilleur outil possible selon notre contexte.

Ensuite, nous avons utilisé les informations de sentiments extraites par l'outil comme paramètre supplémentaire pour entraîner un réseau de neurones récurrent. Ce réseau prenait aussi en entrée des *word embeddings* et des encodages de caractères. L'ajout des informations de sentiment a permis des gains légers, mais statistiquement significatifs, sur l'exactitude, la précision et le rappel lors de la classification de trois jeux de messages toxiques. Nous avons aussi simulé des modifications apportées sur des mots-clés toxiques par un utilisateur malveillant sur les ensembles de tests, et trouvé que les informations de sentiment aidaient encore plus dans ce cas. En effet, l'exactitude avait jusqu'à 3% d'augmentation, principalement causé par le fait que le rappel restait beaucoup plus haut. Ceci est venu confirmé l'hypothèse principale, qui était que le sentiment d'un message était beaucoup plus difficile à dissimuler, et constitue la contribution principale de ce premier article.

Dans le second article, nous nous sommes écartés du traitement de message individuels pour étudier les conversations entières. Utilisant l'outil de détection de sentiment décrit précédemment et les travaux fait par (Zhang et al., 2018), nous avons étudié s'il était possible de prédire si une conversation va dérailler en fonction de ses premiers messages. Tout d'abord, nous avons validé que l'outil de détection de sentiment était bel et bien utile dans ce contexte en vérifiant que les informations de sentiments seules pouvait servir de prédicteur.

Après avoir confirmé la pertinence des informations de sentiments dans ce contexte, nous les avons ajoutés aux caractéristiques développées par (Zhang et al., 2018), puis avons testé le tout sur le même jeux de données. L'ajout du sentiment a permis une augmentation de l'exactitude d'environ 2%. De plus, en analysant l'impact des sentiments dans les premiers messages, nous avons réalisé qu'en prenant un nombre différent d'attributs de sentiments il est possible d'améliorer encore ce résultat. Nous

avons ensuite utilisé la même méthode sur un jeu de données complètement différent et avons obtenu des résultats cohérents, montrant que le modèle peut se généraliser assez bien.

Nous avons aussi observé qu'il est nécessaire de conserver une grande granularité dans les informations de sentiments, et non les combiner en un simple score global. Par exemple, les mots négatifs sont très significatifs, et uniquement quelques uns peuvent rendre une conversation toxique. Une conversation saine doit cependant contenir beaucoup de mots positifs pour être maintenue. Finalement, nous avons aussi noté que lorsque l'on travaille avec des conversations, les quelques messages les plus récents sont les plus importants pour prédire si la conversation peut devenir toxique, car le ton général peut changer très rapidement.

Ces deux articles ne sont que les premiers pas de l'utilisation de sentiment, ou même de d'autres facettes du traitement du langage naturel, pour la détection de toxicité. Il est assez facile de s'imaginer qu'il est possible d'utiliser d'autres caractéristiques en plus des sentiments pour améliorer le traitement de messages individuels. De plus, les travaux sur la détection de toxicité dans les conversations sont encore à leurs débuts, et il reste énormément de travail à faire pour utiliser tout le travail fait sur les messages individuels dans un contexte de conversation. Sans négliger l'importance des résultats que nous avons obtenus, il faut augmenter encore les performances pour bien utiliser cette technologie dans des contextes réels. Il est toutefois clair que l'utilisation de sentiments est une piste prometteuse, et qu'ils seront sans doute importants dans le développement de futurs outils de modération des communautés en ligne.

Bibliographie

Justine Zhang, Jonathan P. Chang, Cristian Danescu-Niculescu-Mizil, Lucas Dixon, Yiqing Hua, Nithum Thain, and Dario Taraborelli. Conversations gone awry : Detecting early signs of conversational failure. *CoRR*, abs/1805.05345, 2018. URL http://arxiv.org/abs/1805.05345.