IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# Abusive Language Detection in Online User Content

Akanksha Tiwari , Er Chandan Kumar , Er. Amitesh Pandit

1 ( M.tech Student)Department of Computer Science And Engineering, IET.Dr. RamManohar Lohia Awadh University Ayodhya(U.P)

2 Assistant Professor, Department of Computer Science And Engineering, IET Dr. Ram Manohar Lohia Awadh university Ayodhya U.P

3 Assistant Professor, Department of Computer Science And Engineering, IET Dr. Ram Manohar Lohia Awadh university Ayodhya U.P

**Abstract**

Detecting abusive language in user-generated online content has become an increasing problem in recent times. years. Most current business methods use blacklists and regular expressions, but these metrics fall in short, in the face of more subtle and less eccentric examples of hate speech. In this work we develop a machine learning-based method for detecting hate speech about the user online feedback from two areas that surpass a cutting edge deep learning approach. We are also developing a body of learning-based method for detecting hate speech about the user online feedback from two areas that surpass a cutting edge deep learning approach. We are also developing a body of user comments marked for abusive language, the first of how nice. Finally, we use our detection tool to analyse abusive language over time and in different contexts to dig deeper into improve our knowledge of this behaviour.

**Keywords:-** Hate Speech, Abusive Language, Stylistic Classification, Discourse Classification

## 1. INTRODUCTION

Every time you engage online, whether it's on the bulletin board forums, comments or social networks, there is always a serious risk of being the target of ridicule and even bullying. Words and phrases like kill myself a $$ hole or they should all burn in hell because what they have done is unfortunately not uncommon online and can have a impact on the courtesy of a community or on a user experience. To combat abusive language, many internet companies have standards and guidelines that users must adhere to and employ human editors, in conjunction with systems that use regular expressions and blacklist, to catch bad language and therefore delete a message. As people communicate more and more online, the need for high quality automated abusive language classifiers gets much deeper. Recent cases highlight the impact of hurtful language on online communities, as well as on large corporations. For example, in 2013 Facebook was criticized for hosting pages that were hateful against women such as Violently raping your friend just for laughs and kick your girlfriend in the fanny because she won't make you a sandwich.

1 Within days, a petition was launched which raised more than 200,000 supporters and several large companies have removed or threatened to remove their ads from Facebook since they were inadvertently placed on these pages. Facebook is not the only company to face these problems; any business that hosts user-generated content will have a moderation problem. This shows the great impact that hate language can have on a community as

well as a large business. On a more individual level, when actor Robin Williams deceased, his daughter Zelda sent him a souvenir father and was immediately bullied on Twitter and Instagram and ultimately deleted all of his online accounts. This harassment prompted Twitter to review and revise its hate speech guidelines.2 While automatic detection of abusive language online is an important subject and task, the state of the art has not been very unified, thus slowing progress. Previous research has lasted different areas ranging from natural language processing (NLP) to web sciences to artificial intelligence, that is to say that several similar methods have been published in the last three years. Plus, abusive language can be a bit of a catch-all term. There are studies [14] that focus on detecting profanity, and others, like [18], that focus on hate speech addressed to a particular ethnic group. To complicate matters further, to date there has been no de facto test with which to compare the methods. In this article, we aim to develop an advanced method
to detect abusive language in user comments, while address the above shortcomings on the ground. More precisely, this article has the following contributions:

• We are developing a supervised classification methodology with NLP features to surpass a deep learning approach. We use and adapt many of the features used in the prior art in order to see how they work on the same data set. We also extend this feature set with features derived from distributional semantic techniques.
• We are releasing a new dataset of several thousand users
comments collected in different areas. This set includes three judgments per comment and for comments qualified as abusive, a more refined approach classification on how each is abusive.

• Previous work was evaluated on a set of fixed and static data.
However, given the problems with changing the language time and also with users trying to smartly escape keyword-based approaches, we do several analyses models trained on different types and sizes of data operate over a period of one, two areas. To our knowledge, this is the first longitudinal study of a computational approach to language detection. In §2, we discuss what abusive language is and what it is so difficult to deal with, as well as the related work. §3 described our datasets and participatory annotation efforts. In §4 we discuss our framework for detecting hate speech and in §5 we discuss a battery of experiments to analyse this tool and hate speech in general. Finally, we want to warn the reader that there are examples of hate speech reproduced in the paper which are here for illustrative purposes only.

## 2.BACKGROUND

### 2.1.Why is this task difficult?

Detecting abusive language is often more difficult than waits for various reasons. data noise in the conjunction with a need for knowledge of the world does not only it is a difficult task to automate but also potentially a difficult task for people too. More than just tagging keywords. Intentional obscuring words and phrases to escape manual or automatic verification often makes detection difficult. Obfuscations such as ni9 9er, whoopiuglyniggerratgolberg and JOOZ make it impossible for simple keyword tracking metrics succeed, especially since there are many permutations for a source word or phrase. Conversely, the use of the keyword spotting could lead to false positives. Difficult to follow all the racial and minority insults. A reasonably effective abuse or profanity classifier can be made with a blacklist (a collection of words known to be hateful or insulting), however, these lists are not static and are constantly evolving. A blacklist should therefore be regularly updated to follow the language change. In addition, some insults that might be unacceptable to a group can be quite good for another group, and therefore the context of the the blacklist word is very important (this forms the motivation for the work of [18]). Abusive language can in fact be very common and grammatical. Although there are many examples on the Internet of very loud abusive language, as in Add another JEW fined a bi $$ ion for flying like a little maggot. Hang thm all., Which can be a useful signal for an automated method, there are actually many instances where abusive language, or even more specifically hate speech, is enough. fluent and grammatical. For example: I am surprised that they reported on this shit who cares about another dead nigger? The abuse can go beyond sentence limits. In the phrase Chuck Hagel will protect Americans from bickering desert animals. Let them kill each other, well riddance !, the second sentence that actually has the most the hateful intensity (they kill each other) depends on the successful resolution of them to desert animals which itself requires knowledge of the world to solve. The point here is that abusive language is not limited to punishment. In certain In this case, the other sentences must be taken into account in deciding whether the text is abusive or contains incidences of hate speech. Sarcasm. Finally, we noted instances where some users post sarcastic comments in the same voice as the people who produced abusive language. It is very difficult for humans or machines to be correct as it requires knowledge

from the community and potentially even from the users themselves: the same thing over and over and over and over again from day to night and day because I am disabled and stay at home. i hate the Jews they ran on my legs with their BMW. so I'm going to blow them up everyday .. i really hurt them i'm so powerful .. if ipost on the Jews here, they all suffer. I sow mighty vwbwbwbwaaahahahahahah I'm crippled but I can destroy them with my posts .. i'm great poster. cccwwbwahahahaha no one can find me .. i'm chicken so i can post behind google wall anonymous posters. ccwbwbwbabahahahah i will give it to her ten thumbs down and slander the Jews .. ccwbwbwbahahahah..i am adoplh hitler reincarnated.

## 2.2 Related Work

Most of the previous work in the area of language abuse detection has in fact been spread over several overlapping areas. This can lead to some confusion as different works may address specific aspects of abusive language, define the term differently or apply it to specific online domains online forums, etc.). To further complicate the comparison between the approaches, almost all previous work uses evaluation sets. One of the contributions of this article is to provide a public dataset in order to better move the field forward. One of the first books to tackle abusive language was[21] who used a supervised classification technique in conjunction with n-gram, manually developed regular expression patterns, contextual characteristics that take into account the abusive nature of the preceding sentences. Since most basic approaches use predefined blacklists, [15] noted that some words on the blacklist might not be abusive in the appropriate context. In their work they have shown a improved detection of profanity using lists as well as an edit distance metric. The latter allowed them to catch non-standard terms such as @ss or sh1t. Another contribution of the work was that they were the first to use crowdsourcing to annotate abusive language. In their task, they used Amazon Mechanical Turk workers to tag 6500 comments on the Internet as abusive or not abusive. Only them used comments in which a majority of the turkers agreed on the label. 9% of the comments were deemed as carrying profane words. In our work, we also make use of crowdsourcing to curate a corpus of several thousand internet comments. The main differences are that we do not limit the task to just profanity and also have the workers annotate for other types of hate speech and abusive language. In addition, we are making this dataset public. [3] was one of the first to use a combination of lexical and parser features to detect offensive language in youtube comments to shield adolescents. While they do note that they do not have a strict definition of offensive language in mind, their tool can be tuned by the use of a threshold which can be set by parents or teachers so online material can be filtered out before it appears on a web browser. The work takes a supervised classification approach using Support Vector Machines (SVMs) with features including n-grams, automatically derived blacklists, manually developed regular expressions and dependency parse features. They achieve a performance on the task of inflammatory sentence detection of precision of 98.24% and recall of 94.34%. One difference between our work and this one is that they attempt to spellcorrect and normalize noisy text before feature extraction. We believe that this noise is a potentially good signal for abuse detection and thus have features to capture different types of noise. Our work also makes use of dependency features, though with a much broader set of tuples than [3]. [18] provide the most comprehensive investigation of hate speech (hateful language directed towards a minority or disadvantaged group) to date, with working definitions and an annotation task. Here their focus was less on abusive language and more specifically on anti-Semitic hate. First, they manually annotated a corpus of websites and user comments, with Fleiss kappa interlobular agreement at 0.63. Next, they adopted a related approach to the aforementioned supervised classification methods by first targeting certain words that could either be hateful or not, and then using Word Sense Disambiguation techniques [20] to determine the polarity of the word. Their method performs at 0.63 F-score. To our knowledge, this is the only work to target hate speech and the only one to have done a rigorous annotation of data, though the set could not be made public. We build on their work by crowdsourcing the annotation of a data set of user comments, categorizing each comment as abuse, profanity, and/or hate speech. This set will be made public. Finally, [5] use a paragraph2vec approach adopted from [8] to classify language on user comments as abusive or clean. Their approach outperformed a bag-of-words (BOW) implementation (0.8007 to 0.7889 AUC). In our work, we use a more sophisticated algorithm to learn the representation of comments as low-dimensional dense vectors. Moreover, our representation is learned using only unigrams in order to compliment other relevant features. In our work, we aim for a method that is efficient and flexible but also operates at a high accuracy by combining different light-weight features. We include an evaluation using their data to directly compare our system but also experiment with their approach as additional features in our methodology.

# 3.DATA

All data used for training and testing in this article was taken from reviews found on Yahoo! Finance and news. These comments were moderated by Yahoo employees including the main function was to provide editorial labels for various annotation / editorial tasks. All subjects had at least an undergraduate degree and were familiar with the concept of judge text passages for different types of annotation tasks and the requirements. Before taking the actual moderation task, they have been trained to familiarize themselves with with text judgment guidelines. As mentioned in §2.2, there are many forms of abusive language, and sometimes the term abusive language is confused with hate speech. For our work, abusive language encompasses hate speech, blasphemy and contempt. Language. A summary of guidelines and examples for each category is shown in Table 4.

## 3.1 Primary Data Set

Data for our primary training models is sampled from comments posted on Yahoo! Finance and News during the period between October 2012 and January 2014. The original data is collected as follows. A completely random 10% subset of the comments that are posted each day on Yahoo! Finance and News articles are sent for review by Yahoo's in-house trained raters. Further, all comments which are reported as "abusive" for any reason by visitors to the Finance and News portals are also sent to the raters for review and judgment. Such reports are termed community moderation. To maintain business confidentiality, we cannot divulge the volume of comments in the community moderation bucket. The breakdown of "Clean" and "Abusive" comments for both domains is shown in Table 1. The percentage of abusive comments for Finance is roughly 7.0% for Finance and 16.4% for News. In our experiments with this set, we train on 80% of the data and test on the remaining 20%.

**Table 1: Primary Data Set Statistics**

| Finance data | | News data | |
|---|---|---|---|
| Clean | 705,886 | Clean | 1,162,655 |
| Abusive | 53,516 | Abusive | 228,119 |
| Total | 759,402 | Total | 1,390,774 |

## 3.2 Temporal Data Set

The data used in our temporal experiments (§5.4) are also sampled from comments posted on Yahoo! Finance and News; the period is between April 2014 and April 2015. The number of "Clean" and "Abusive" comments on both domains are shown in Table 2. The ratio of abusive comments on each domain is lower than our older main data, around 3.4% on Finance, 10.7% on News. The main reason is our collective efforts including deploying our models in the production environment during that time.

**Table 2: Temporal Data Set Statistics**

| Finance data | | News data | |
|---|---|---|---|
| Clean | 433,255 | Clean | 655,732 |
| Abusive | 15,181 | Abusive | 70,311 |
| Total | 448,436 | Total | 726,073 |

## 3.3 WWW2015 dataset

To compare directly with previous work, we also use the dataset described in [5]. The set includes 56,280 comments labeled "abusive" and 895,546 comments labeled like "Clean" collected on Yahoo! Fund in the same way as dataset 1. The percentage of abusive comments on this data is about 5.9%. To reproduce their assessment methodology, we carry out a cross-validation in 5 times on this set.

Table 3: WWW2015 Statistics

| | |
|---|---|
| Clean | 895,456 |
| Abusive | 56,280 |
| Total | 951,736 |

## 3.4 Assessment data set

In addition to the previous three datasets, we wanted a review-specific corpus in which each comment is tagged by three trained reviewers. This overlap labeling allows us to determine the level of human agreement in this task. We extracted several thousand comments between March and April 2015 for reviewers to label. For the "Clean" and binary distinction "Abuse", the concordance rate is 0.922 and Fleiss's Kappa is 0.843, which shows that humans can reach a relatively high agreement for this task. We have also Did the reviewers tag the abuse subcategory (hate, derogatory language, and profanity), where multiple subcategories can be tagged for comment. The agreement for this task falls at 0.603 and Fleiss's Kappa at 0.456. From that tagged set we used 1,000 tagged as "Clean" and 1,000 marked as "abusive" for a total of 2,000 comments. We used the majority vote of this binary categorization to evaluate our model in section §5.3. We will do this public data via the Yahoo Webscope program. 3 To our knowledge that this is the first publicly available dataset on abusive languages with multiple annotations and different subcategories. There is a data.

## 4. METHODOLOGY

For our work we use a supervised classification method which uses NLP features that measure different aspects of the user's comment. Specifically, we use the Vowpal Wabbit
regression model5 in its standard setting with a bit rate of28. We are basing our NLP features on previous work in the field of feelings[9], text standardization [1] among others. Our features can be divided into four classes: N-grams, linguistics, syntax and distributional semantics. For the first three features, we do a gentle pretreatment to transform some of the noise found in the data that could impact the number of sparse elements of the model. Examples of transformations include normalizing numbers, replacing long unknown words with the same token, replacing repeated punctuation with the same token, etc. For the fourth class of objects, we did none of the above normalisation.

## 4.1 Characteristics of N-gram

We use n-gram characters (3 to 5 characters, spaces included) and symbolic unigrams and bigrams. Unlike previous work in this area which ignored non-normalized text or used simple edit distance measures to normalize them, we use n-grams of characters to model types of conscious or unconscious bastardises of offensive words.

• number of one letter tokens
• number of capitalized letters
• number of URLS
• number of tokens with non-alpha characters in the middle
• number of discourse connectives, based on
• number of politeness words
• number of modal words (to measure hedging and confidence by speaker)
• number of unknown words as compared to a dictionary of English words (meant to measure uniqueness and any misspellings)
• number of insult and hate blacklist words

## 4.2 Linguistic characteristics

To better manage data noise, we have developed specialized features based on the work of [1]. These features are intended to explicitly search for inflammatory words (such such as the use of pre-existing hate lists) but also elements non-abusive language such as the use of polite words or modal verbs. These features include:

• comment length in tokens
• average word length
• number of punctuations
• number of periods, question marks, quotation marks and repeated punctuation

**Table 4: Annotation Instructions**

| | |
|---|---|
| Given the passage below, first rate the passage on whether the passage is clean or abusive. And if abusive,<br>• whether it contains hate speech<br>• whether it contains derogatory languge<br>• whether it contains profanity | |
| **Hate Speech**: | language which attacks or demeans a group based on race, ethnic origin, religion, disability, gender, age, disability, or sexual orientation/gender identity. Some examples:<br>• The JEWZ are Market Manipulators Hate them freaking Jews !! To bad Hitler did not finish them off !!!<br>• I hate the mall because there are too many unemployed apefricans and mudsharks.<br>• 300 missiles are cool! Love to see um launched into TelAviv! Kill all the gays there!<br>• EGYPTIANS ARE DESCENDED FROM APES AND PIGS. NUKE EGYPT NOW Ah me tu se ku sh |
| **Derogatory**: | language which attacks an individual or a group, but which is not hate speech. Some examples:<br>• All you perverts (other than me) who posted today, needs to leave the O Board. Dfasdfdasfadfs<br>• yikes...another republiCUNT weighs in.... |
| **Profanity**: | language which contains sexual remarks or profanity. Some examples:<br>• T.Boone Pickens needs to take a minimum wage job in FL for a week. I guarantee he shuts the f up after that.<br>• Somebody told me that Little Debbie likes to take it up the A.$.$.<br>• So if the pre market is any indication Kind of like the bloody red tampons that you to suck on all day?? |

• number of tokens of a letter
• number of uppercase letters
• number of URLs
• number of tokens with non-alpha characters in the middle
• number of speech connectors, based on
• number of polite words
• number of modal words (to measure coverage and confidence per speaker)
• number of unknown words compared to a dictionary of English words (intended to measure uniqueness and spelling mistakes)
• number of insulting and hateful words on the blacklist6

## 4.3 Syntax characteristics

The use of natural language analysis is common for tasks ranging from sentiment analysis [9] to best response prediction in the CQA analysis [16]. We derive functionality from ClearNLP dependency analyzer v2.07. The features are basically different types of tuples using words, POS tags and dependency relationships. These include:
• parent of the node
• grandparent of the node
• Parent's POS
• Grandparents' POS
• tuple composed of the word, the parent and the grandparent
• children of node8 tuples made up of permutations of the word or its POS, the dependency label connecting the word to his parent, and the parent or his PDV The motivation behind these features is to capture long range dependencies between words that n-grams may not be able to do (as in the example: Jews are lower class pigs, where an n-gram model could not connect Jews and pigs, but the use of an addiction analyzer generate the tuple -are-Jews-pigs where are Jews and pigs the children of are.

## 4.4 Characteristics of distributional semantics

The ideas of the word and the text distributed and distributed representations has supported many applications in successful language processing. The related work is largely focused on the notion of representations of words and texts (as in[10], [8] and [4]), which improve previous modeling efforts Lexical semantics using vector space models [10]. Nowadays, only [5] used it in their approach to violence language detection. We use three types of features derived from embedding. the the first two are based on the average of all word embeddings words in the commentary, in essence a superficial method meant to approximate an embedding for a larger piece of text that has had some success in tasks such as sentiment analysis[6]. In one, we use pre-trained9 embeddings derived from a large current text corpus (now pre-trained. In the second we use word2vec10 to form the incorporations from our large corpora of financial information and commentary respectively (now word2vec). For both functionalities, we use a 200-dimensional embedding vector. More recently, the concept of embedding has been extended beyond words to a number of text segments, including sentences [11], sentences and paragraphs [8], entities [19] and

documents. For our third integration functionalities, we develop a comment integration approach similar to [8]. In order to get the comment overlays we learn distributed representations for our comment dataset. the the comments are represented as vectors of low dimension and are learned in conjunction with distributed vector representations of tokens using a distributed memory model explained in [8]. In in particular, we take advantage of the content of comments to

model the sequences of words within them. While the word vectors help predict the next word in the comments, comment vectors also help to predict the next word given many contexts sampled from the comment. In our comments integration model (now comment2vec, each comment is mapped to a unique vector in a matrix representing the comments and each word is mapped to a unique vector in a matrix representing words. Then the comment vectors and the word vectors are concatenated [8] to predict the next word in a context. More precisely, the probability distribution of observing a word does not depend only on the number of surrounding words, but also depends on the specific comment. In this way, we represent each comment by a low dimensional dense vector which is formed to predict words in the commentary and overcomes the weaknesses of the word embeddings only. We form the integration of words in comments using skip-bigram model [10] with a window size of 10 using a softmax hierarchical drive. For the integration of comments, we leverage the distributed memory model as it generally works well for most tasks [8]. We form a low dimension model (100 dimensions) as we intend to add these representations to other features such as n-gram distributions. We also limit the number of iterations to 10 to increase the Efficiency. A great advantage of learning distributed representation vectors of comments in this way is that the algorithm is is not sensitive to the length of comments and does not require specific adjustment for the weight of the words. As a downside, this algorithm needs constant recycling when new comments are added, making the model less efficient for online applications. This challenge can be met in different ways:

i) scalable vector tuning and update for new comments,

ii) derive a low dimensional vector for the new comments using gradient descent using parameters, word vectors and the softmax weights of the trained model, and iii) approximation of the new vector by estimating the distance of the new comment to previous comments using the words and their representations. We plan to study these methods in future works.

## 5. EXPERIMENTS

In this section, we describe a battery of experiments meaning to evaluate our classifier, compare it to previous work then use it as a tool to analyze trends in user hate speech comments. In §5.1 we show the overall performance of our model on Primary Finance and News datasets. We evaluate the impact of each feature and discuss what is best for it task. In §5.2 we then compare our model to the previous work of[5] on the WWW2015 set. Then we evaluate on our curated Evaluation data set (§5.3) and in §5.4 we study the question: how does performance vary over time? We could hypothesize that the language and use of hate speech is changing quickly and this will therefore have an impact on the performance of a classifier if the model is not updated.

### 5.1 Assessment on the primary data set

In this set of experiments, we train and test our model using the main dataset for the two domains (Finance and News). For each area, we use 80% for training and 20% to test. Table 5 shows the results for each domain when a model trained with a single type of entity as well as with all combined characteristics. For both areas, combining all functionalities gives the best performance (0.795 for finance and 0.817 for news). News has a slight performance advantage can be easily explained by the fact that there is a training corpus available for this area. In terms of individual characteristics, for both sets, character ngrams have the greatest contribution. Both sets exhibit different behavior in terms of other features. In finance set, the syntactic and distributional semantic functionalities do not work as well as in the news area. we thinks that Finance is slightly noisier than News and therefore these more complex features are not doing as well

Table 5: Primary Data Set Results (by F-score)

| Features | Finance | News |
|---|---|---|
| Lexicon | 0.539 | 0.522 |
| Trained Lexicon | 0.656 | 0.669 |
| Linguistic | 0.558 | 0.601 |
| Token N-grams | 0.722 | 0.740 |
| Character N-grams | 0.726 | 0.769 |
| Syntactic | 0.689 | 0.748 |
| word2vec | 0.653 | 0.698 |
| pretrained | 0.602 | 0.649 |
| comment2vec | 0.680 | 0.758 |
| **All Features** | **0.795** | **0.817** |

## 5.2 Overall assessment WWW2015

We then conducted an experiment on the data used in[5] to directly compare our work. As in their experimental configuration, we use 5-fold cross validation. Table 6 shows that our the model surpasses the state of the art by 10 AUC points (0.9055 to0.8007). We also report the precision, recall and F score for our model with all the features and several other baselines. Simply use our blacklist lexicon (lexicon) as a lookup table produces an F score of 0.537. Train a model on this lexicon so that we have weights on each word produces a a slightly better F score of 0.595. As in §5.1, the token and The n-grams of characters themselves are extremely predictive and surpass [5]. They are also extremely close in terms of performance to our model with all the features. The distributive features don't perform as well as their n-gram counterparts but easily surpass lexicon-based baselines. They too help improve overall results by increasing recall precision. Among the distribution features, comment2vec surpasses word2vec mainly because the comment2vec algorithm preserves the semantic aspect of taking comments advantage of co-formation of words and comments. On the other hand, simply using average word representations reduces sensitivity to context and word order and possibly semantics of comment embedding.

Table 6: WWW2015 Results

| Method | Rec. | Prec. | F-score | AUC |
|---|---|---|---|---|
| [5] | - | - | - | 0.8007 |
| Lexicon | 0.557 | 0.519 | 0.537 | - |
| Trained Lexicon | 0.540 | 0.662 | 0.595 | 0.7597 |
| Linguistic | 0.501 | 0.523 | 0.512 | 0.6463 |
| Token Ngrams | 0.771 | 0.713 | 0.741 | 0.8532 |
| Character Ngrams | 0.821 | 0.732 | 0.774 | 0.9037 |
| Syntactic | 0.723 | 0.593 | 0.651 | 0.7902 |
| word2vec | 0.766 | 0.597 | 0.671 | 0.8409 |
| pretrained | 0.714 | 0.566 | 0.631 | 0.7851 |
| comment2vec | 0.780 | 0.590 | 0.672 | 0.8521 |
| **All Features** | **0.794** | **0.773** | **0.783** | **0.9055** |

## 5.3 Assessment dataset experience

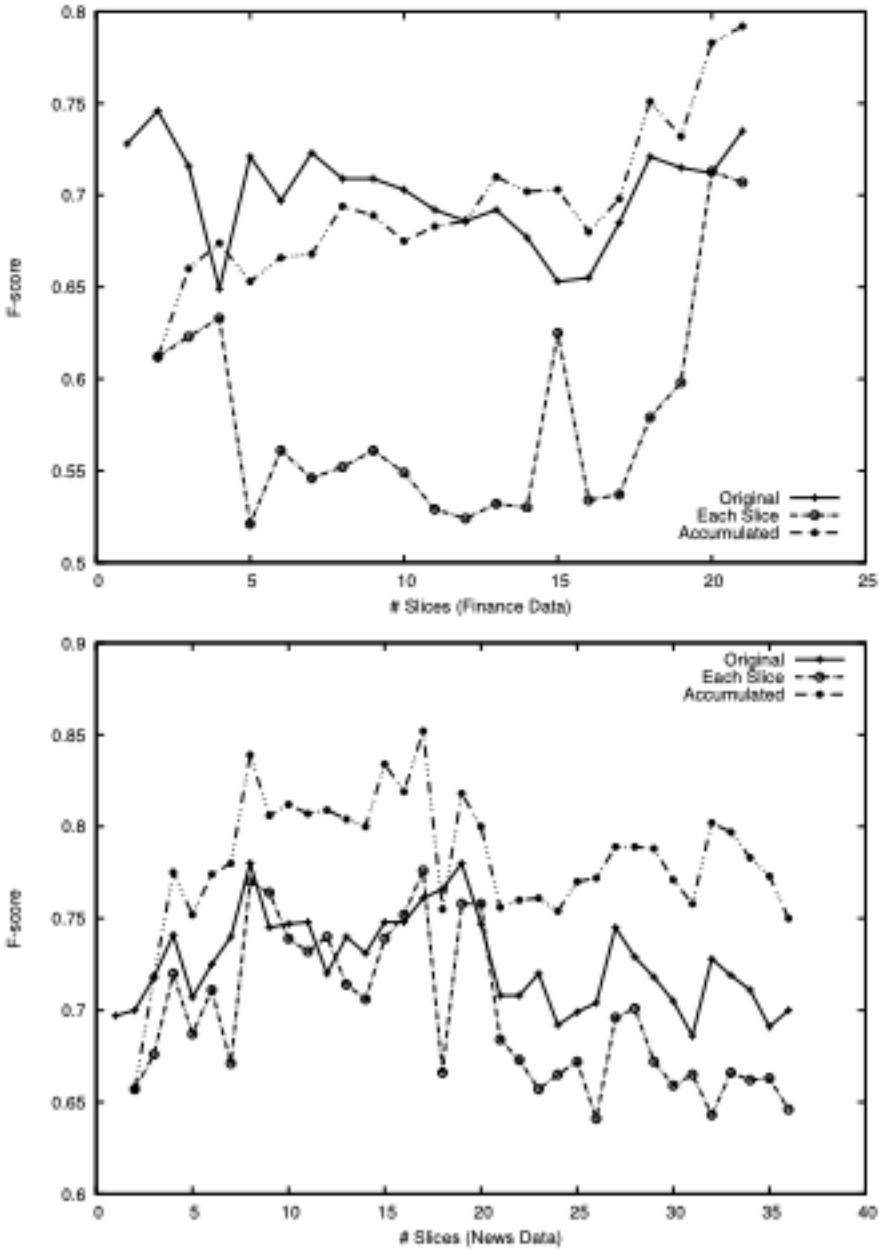
We applied our model to the more highly organized assessment dataset discussed in §3.4. The ground truth labels are obtained as majority vote of judgments attributed manually. The results of the evaluation of our models on these data are presented in table 7. The performance is comparable to our results in §5.1 and §5.2. In addition, we have experimented with different gold standard references: we evaluated the system when all three evaluators agree (unanimous agreement) and where exactly two agreed (and we use their judgment). Although the number of comments "All accepted" is dominant this data (1,766 out of 2,000) and the difference between the labels by majority vote and those by "All agree" are small, the results on cases where all evaluators agree results compared to those with exactly 2 out of 3 reviewers agreeing(0.839 to 0.826). Some of the false positive cases are questionable eg. our"abusive" template tags for comments such as "Gays in Indiana is pooping her pants because of this law. "Bug is a lie for the fear-mongering scammers of the MM Warmie sect." stop the black on white crimes! '(Sic), which are labeled as "Clean" by 2 out of 3 reviewers. Some comments are also inherently ambiguous, eg. "Soak their clothes in gasoline and put them on in fire. "," Or you could ... you know ... shoot them "could be"Abusive", but without the context it is difficult to judge. Part of our future job is to extract the comment thread and use them as context to judge each comment.

**Table 7: Evaluation Data Set Evaluation**

| Experiment | n | Recall | Precision | F-score |
|---|---|---|---|---|
| Majority | 2,000 | 0.825 | 0.827 | 0.826 |
| All Agreed | 1,766 | 0.842 | 0.837 | 0.839 |
| 2 of 3 Agreed | 234 | 0.378 | 0.500 | 0.431 |

## 6. CONCLUSIONS AND FUTURE WORK

As the amount of user-generated content online quickly grows, it is necessary to use precise and automated methods reporting abusive language is of utmost importance. do not Resolving the issue may lead to users leaving an online community due to harassment or companies pulling ads that appear alongside abusive comments. Although there has been a lot of work in this area in several different related areas, to date there has been no standard evaluation set with which researchers could compare their methods. In addition, several NLP methods have been used in previous work, but these features have never been combined or evaluated against each other. In our work we take a big step forward in the field by first providing an organized public dataset and also performing several assessments a range of NLP features. We have experimented with several new features for this task: different syntax features as well as different types of integration features, and find them very powerful when combined with standard PNL features. Single character ngrams do very well in these loud data sets. Our model also outperforms a deep learning-based model while avoid the problem of having to recycle the embeds oneach iteration. Then we used our model to perform a analysis of hate speech over a period of one year, providing practical information on the amount of data and the type of data is required for this task. So far, most of the work has focused on abuse in English but it remains to be seen how our approach or any of the other earlier approaches would be accepted in other languages. Given the power of the two n-gram functions in English, these would probably do well in other languages given enough training data. Another area of future work is to use the context commentary as additional features. The context could include the article to which it refers, any preceding comments or replied, as well as information about the commentator past behavior or comments




# 7. REFERENCES

[1] S. Brody and N. Diakopoulos. Cooooooooooooooollllllllllllll!!!!!!!!!!!!!! using word lengthening to detect sentiment in microblogs. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 562–570, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

[2] M. D. Buhrmester, T. Kwang, and S. D. Gosling. Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? Perspectives on Psychological Science, 6(1):3–5, Jan 2011.

[3] Y. Chen, Y. Zhou, S. Zhu, and H. Xu. Detecting offensive language in social media to protect adolescent online safety. In Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom), pages 71–80. IEEE, 2012.

[4] N. Djuric, H. Wu, V. Radosavljevic, M. Grbovic, and N. Bhamidipati. Hierarchical neural language models for joint representation of streaming documents and their content. In International World Wide Web Conference (WWW), 2015.

[5] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati. Hate speech detection with comment embeddings. In Proceedings of International World Wide Web Conference (WWW), 2015.

[6] M. Faruqui and C. Dyer. Non-distributional word vector representations. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 464–469, Beijing, China, July 2015. Association for Computational Linguistics
.

[7] J. Horton, D. G. Rand, and R. J. Zeckhauser. The online laboratory: Conducting experiments in a real labor market. National Bureau of Economic Research Cambridge, Mass., USA, 2010.

[8] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In T. Jebara and E. P. Xing, editors, Proceedings of the 31st International Conference on Machine Learning (ICML-14), pages 1188–1196. JMLR Workshop and Conference Proceedings, 2014.

[9] B. Liu. Sentiment Analysis and Opinion Mining. Morgan Claypool Publishers, 2012.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781, 2013.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words andphrases and their compositionality. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, Advances in Neural Information Processing Systems 26, pages 3111–3119. Curran Associates, Inc., 2013.

[12] G. Paolacci, J. Chandler, and P. G. Ipeirotis. Running experiments on amazon mechanical turk. Judgment and Decision Making, 5(5):411–419, 2010.

[13] E. Pitler and A. Nenkova. Using syntax to disambiguate explicit discourse connectives in text. In Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pages 13–16, Suntec, Singapore, August 2009. Association for Computational Linguistics.

[14] S. Sood, J. Antin, and E. Churchill. Profanity use in online communities. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 1481–1490. ACM, 2012.

[15] S. O. Sood, J. Antin, and E. F. Churchill. Using crowdsourcing to improve profanity detection. In AAAI Spring Symposium: Wisdom of the Crowd, 2012.

[16] M. Surdeanu, M. Ciaramita, and H. Zaragoza. Learning to rank answers to non-factoid questions from web collections. Computational Linguistics, 37:351–383, 2011.

[17] S. Suri and D. J. Watts. Cooperation and contagion in web-based, networked public goods experiments. PloS One, 6(3), 2011.

[18] W. Warner and J. Hirschberg. Detecting hate speech on the world wide web. In Proceedings of the Second Workshop on Language in Social Media, pages 19–26, Montr´eal, Canada, June 2012. Association for Computational Linguistics.

[19] B. Yang, W. Yih, X. He, J. Gao, and L. Deng. Embedding entities and relations for learning and inference in knowledge bases. CoRR, abs/1412.6575, 2014.

[20] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In Proceedings of the 33rd annual meeting on Association for Computational Linguistics, pages 189–196. Association for Computational Linguistics, 1995.

[21] D. Yin, Z. Xue, L. Hong, B. D. Davison, A. Kontostathis, and L. Edwards. Detection of harassment on web 2.0. Proceedings of the Content Analysis in the WEB, 2:1–7, 2009.