On Online Hate Speech Detection. Effects of Negated Data Construction

Cheniki Abderrouaf
University of Oulu, Faculty of Information Technology, CMVS
PO Box 4500, Oulu 90014 FINLAND
cheniki.Abderrouaf@student.oulu.fi

Mourad Oussalah
University of Oulu, Faculty of Information Technology, CMVS
PO Box 4500, Oulu 90014 FINLAND
mourad.oussalah@oulu.fi

***Abstract*— In the era of social media and mobile internet, the design of automatic tools for online detection of hate speech and/or abusive language becomes crucial for society and community empowerment. Nowadays of current technology in this respect is still limited and many service providers are still relying on the manual check. This paper aims to advance in this topic by leveraging novel natural language processing, machine learning, and feature engineering techniques. The proposed approach advocates a classification-like technique that makes use of a special data design procedure. The latter enforces a balanced training scheme by exploring the negativity of the original dataset. This generates new transfer learning paradigms, Two classification schemes using convolution neural network and LSTN architecture that use FastText embeddings as input features are contrasted with baseline models constituted of Logistic regression and Naives' Bayes classifiers. Wikipedia Comment dataset constituted of Personal Attack, Aggression and Toxicity data are employed to test the validity and usefulness of the proposal.**

Keywords—Hate speech, NLP, text mining

I. INTRODUCTION

With the exponential increase of user-generated web content, especially, on social media platforms, the amount of hate speech is also dangerously increasing, which, raises serious challenges to community, policy-makers, and scientists. Such content can alienate users and provide support for radicalization and incitation to violence (Allan, 2013). In this regard, hate speech is defined as any communication that expresses hatred of a group in society and disparages a person and/or group on the basis of some characteristics such as race, color, ethnicity (Nockleby, 2000).

Platform operators, e.g., Facebook, Twitter, Reddit, Riot Games attempted to discourage users from using hate-speech through a combination of policy and platform alteration. Current solutions rely mainly on user reporting of objectionable content, which, in turn, requires labor-intensive review by platform staff members, or through third-party entities such as Amazon Truck. Similarly, some internet companies put forward standards and guidelines that users must adhere to and employ human editors through the blacklisted systems for catching bad language and take appropriate action. For instance, both Facebook and Twitter have responded to criticism for not doing enough to prevent hate speech on their sites by instituting policies to prohibit the use of their platforms for attacks on people based on characteristics like race, ethnicity, gender, and sexual orientation, or threats of violence towards others[1]. On the other hand, several national and international initiatives have been launched over the past few years to increase children's online safety. Examples include KiVa, a Finnish cyberbullying prevention programme (http://www.kivaprogram.net/), the 'Non auharcelement' campaign in France, Belgian governmental initiatives and helplines (e.g.clicksafe.be, mediawijs.be) that provide information about online safety. Nevertheless, the result is far from satisfactory because it relies on a manual annotation to identify and delete offensive materials, which makes the process labor-intensive, time-consuming and neither sustainable nor scalable (Salawu et al., 2017). Therefore, yet reliable solutions for online hateful speech are lacking, which created tremendous challenges for both operators and society. For instance, Facebook came under fire for hosting pages which were using hateful and violent words against women like "raping your friend just for laughs"[2] and "kicking your girlfriend in the fanny due to not making a sandwich"[3] in 2013. Besides, several major companies either pulled or threatened to remove their ads from Facebook because a petition was started which amassed over 200000 supporters in just under one day due to hateful content. This puts a lot of pressure on companies hosting user-generated content. Current research into automatic detection of abusive language in online platforms focused mainly on keyword-based search, combined with some semantic content analysis techniques, or machine-learning based strategy on selected corpus, while utilizing techniques from natural language processing (NLP) (Chen et al., 2012;. Kwok and Wang, 2013; Gamback and Sikdar, 2017). In this respect, Warner and Hirschberg (2012) were probably among the early researchers to use machine-learning based classifiers for

[1] Facebook's policy can be found here: www.facebook.com/communitystandards#hate-speech. Twitter's policy can be found here: support.twitter.com/articles/ 20175050

[2] https://sswmen.wordpress.com/2013/07/11/violently-raping-your-friend/

[3] https://www.inquisitr.com/679544/facebook-ads-rape/

detecting abusive language. Nobata et al. (2016) combined pre-defined language elements and word embedding to train a regression model. Djuric et al., (2015) proposed a word-embedding based representation.

Alternative or complementary to machine learning like approaches, one also notices advances in syntactic and rule-based reasoning e.g., Reynolds et al., 2011; Foong and Oussalah, 2017). Burnap and Williams (2015) developed a rule-based approach to classifying antagonistic content on Twitter using associational terms as well as *accusational* and attributional terms targeted at a person or persons following a socially disruptive event as features. Chen et al. (2012) identified offensive content by using both a set of ad-hoc rules that model offensive content and a set of features constituted of profanities, obscenities, and pejorative terms, weighted accordingly based on the associated strength of the term, as well as references to people. Their results showed an improvement in standard machine learning approaches in terms of a false negative rate.

Spertes (1997) suggested a set of syntactic constructs based on word positioning that best describes insulting or condescending in his proposed "Smokey" abusive message classification tool. Mahmud et al. (2008) followed a similar approach but also incorporated relationships between terms to identify "flaming" behavior online. The identification of syntactic relationships within the text is possible via the development of parsing tools such as the Typed Dependency parser from Stanford (Marneffe et al., 2006).

On the other hand, research in machine learning-based classification has also considered the number of classes in the classifier, e.g., abusive versus non-abusive classification as in a binary-classification, or distinguishing among various types of abusive speech. For instance, Zhang and Luo (2018) distinguished abusive language related to racism and sexism together with non-abusive language as well.

Nevertheless, scrutinizing the vast literature of abusive language detection, one acknowledges the dominance of machine learning-based approaches, with their inherent difficulty to scale up the nature, type and complexity of the abusive language structure. Indeed, the key difficulties encountered by the above approaches rely on (i) the challenges associated with the definition of hate speech discourse, where the presence of a wording insult, for instance, does not necessarily entail a hate speech post, and (ii) the limited scope of training samples, which questions the effectiveness of any machine learning like approach due to the constant evolving of hate-speech corpus and the variety of expressions therein. The research has also highlighted that many of these approaches are largely biased towards detecting content that is non-hate as opposed to detecting and discriminating real hate hateful content, possibly, because the non-hate content may not contain any discriminating features (Burnap and Williams, 2016). This motivates the current work, which aims to contribute to the lack of scalability and large bias observed in non-hate speech detection. For this purpose, this paper advocates two key novelties:

1) We investigate a special refinement of textual posts through reshaping the negation connectives in the post. This is motivated by the fact that hate speech can be substantially turned up or down through a simple introduction or removal of the corresponding negation token. It is therefore interesting to evaluate the extent to which the negation connective can influence the performance of the hate-speech detection algorithm.
2) We investigate a transfer learning scheme that allows us to test the inter-operability of pre-trained models from a given hate-speech corpus to another one, in a way to enrich the detection capability and reduce the requirement of large scale training dataset for each case study.
3) Finally, a set of comparisons with some state of the art machine learning models and publicly available datasets has been carried out to demonstrate the feasibility and high performance of the developed approach.

The rest of this paper is organized as follows. Section 2 highlights some of the state of art in terms of hate-speech detection and negation handling. Section 3 details the negation handling proposed approach. Evaluation results are presented in Section 4. Finally, conclusions are presented in Section 5.

II. State of art and background

A. Hate and abuse language definion

Ultimately, the scope of hate and abusive language is quite broad and their definitions were sometimes disparate given the variety of teams who are interested in this matter, ranging from legal entities, economic, social to computational linguistics and information processing communities. For instance, the European Court of Human Right (ECHR) does not offer a specific definition for "hate speech" but instead offers only a set of parameters by which prosecutors can decide if the "hate-speech" is entitled to the protection of freedom of speech.

Especially, hate-speech has been linked to attacks of a person or a group on the basis of protected attributes such as race, religion, ethnic, origin, national origin sex, disability, sex orientation or gender identity (Nockleby, 2000).

Throughout this paper, we deliberately avoid the inherent separation and implicit differences in definitions of hate-speech, abusive language, and offensive language.

We assume that any of hate-speech, abusive language, and offensive language involve a target group constituted of persons or community, and express hatred towards the targeted group or intend to humiliate/insult any members of that group. In other words, this involves an action in which a user intentionally annoys one or more other users in a web community.

B. Feature Set

Central to any "message content"-based research for abusive/hate language detection is the choice of features used as inputs for either machine learning model or rule-based strategy. In this respect, one distinguishes several features depending on whether textual or metadata information is employed (Manning and Shutze, 1999).

IF-IDF features

The approach represents each post as a vector of terms and each term is represented in the vector by its TF-IDF value.

Terms that appear in the corpus but not in a given post will receive a Zero-weight TF-IDF value.
More specifically, the weight of term i in post j is:

$$TFIDF_{ij} = TF_{ij}.IDF_i$$

with

$$TF_{ij} = \frac{n_{ij}}{\sum_k n_{kj}}$$

where n_{ij} is the number of occurrences of term i in post j, and the denominator is the count of the occurrences of all terms in post j.

$$IDF_i = \log \frac{|P|}{|p_j : t_i \in p_j|}$$

where |P| stands for the total number of posts in the whole dataset, $|p_j : t_i \in p_j|$ is the number of posts in which the term t_i appears.
Especially, term frequency provides a measure of how important a particular term is in a given post (a local weighting). While, the inverse document frequency provides a measure of how important a particular term is within the entire corpus (a global weighting). IDF scores are higher for terms which are good discriminators between posts (i.e., terms appearing in many posts will receive lower IDF score)

N-gram features
Traditional n-grams are sequences of n elements (where n is often less than five) as they appear in texts. These elements can be words, characters, POS tags, or any other elements as they encounter one after another in texts. Independence assumption is made such that each word depends only on the last n-1 words.
Typically, the set of n-grams is basically generated by moving a window of n words along the document under consideration, one word at each time. Then, the number of occurrence of each n-gram is counted. A key advantage of such feature is that we do not need to perform advanced segmentation, neither to adopt a dictionary or language specific technique, but, on the other hand, for large corpus, the number of n-grams becomes extremely huge, and many will have no discrimination power.

Linguistic features
These features are intended to explicitly look for various stylistic, capitalization, statistics on wording / post length, usage of politeness words, modal words, hate / swear words, personal pronoun, repetition, among others. Especially, Linguistic Inquiry and Word Count (LWIC) (Pennebaker et al., 2015), among other lexical, contains already numerous word categories associated to aggression, anger, sadness, positive/negative emotion that can be utilized. In short, linguistic features may include: i) length of comment in tokens; ii) average length of word; iii) number of punctuations; iv) number of question marks, quotes, and repeated punctuation; v) number of capitalized letters, URLS, repeated letters /words; vi) number of politeness words; vi) number of modal words; vii) number of insult and hate blacklist words; viii) second-person pronouns, e.g., you, your, yourself; ix) personal pronouns.
A personal pronoun appearing near profanity, for instance, is a good indicator of harassment.

Syntactic features
This captures the inherent relationship that exists between distinct words on the same post. Especially, using freely available dependency parsers, syntactically related word pairs can be inferred, and provide a more complete document description than bigram or trigram like representation. The parser output shows the dependency related words, their parts of speech, and the syntactic relation between them.
The advantage of such features is to capture long-range dependencies between words which n-grams may not be able to do. (such as in the example: Jews are lower class pigs, where an n-gram model would not be able to connect Jews and pigs, however, using a dependency parser would generate the tuple - are-Jews-pigs where Jews and pigs are the children of are.

Distributional semantic features
Distributional semantics, also known as vector space semantics, is an empirical method for the analysis of lexical meaning (Lenci, 2018). It is a usage-based model of meaning, based on the assumption that the statistical distribution of linguistic items in context plays a key role in characterizing their semantic behavior. Distributional representations are built from text corpora as samples of language usage and offer new ways to investigate the interplay between meaning and contexts, and to tackle the dynamicity and plasticity of meaning.
Especially, word embeddings represent the meaning of tokens as a low dimensional real-valued vector; typically, between 100 and 300 dimensions such that words which convey similar meanings or appear in similar contexts would have similar vector based representations.
The development of word2vec and its various refinements, where each word is represented as a vector in 300 dimensions provided a tremendous boost to the development of distributional semantic features. The question of extending the embedding from word level to sentence, post or paragraph level has been approached from different perspectives. A typical solution consists of averaging the word embeddings (from some pre-trained model) of all words in the post/sentence/paragraph. Nevertheless, the development of other embedding mechanisms that account for a post, phrase or sentence, e.g., paragraph2vec, document2vec representations, as well as learning distributional representations for the underlying set of posts.

C. *Transfer Learning*

Transfer learning is the idea of utilizing features, weights, or otherwise defined knowledge acquired for one task to solve another related problem. This makes it useful in situations where only limited data is available for training, validation, and testing (Pan and Yang 2010). This bears analogy to the concept of multi-task learning, which aims to train a model that generates outputs for several related tasks from a single common input, utilizing the similarities and subtle differences in annotations and datasets to improve performance on and regularize against another. Therefore, transfer learning aims to compensate for lack of training data

by adapting supervised models trained in a resource-rich context to a resource scare-context, contributing to enhancing the overall performance of the underlying classifier.

Interestingly, transfer learning also accounts for cases where there is a lack of compatibility of context. For instance, the word "moron" is universally abusive in all domains and would generalize well, while the word "fruit" is almost always completely non-abusive except in specific domains where it might denote a derogatory slang for a homosexual person. Therefore, accounting for such domain-specific would ultimately boost the detection capabilities of hate /abusive speech.

More formally, transfer learning involves the concepts of domains and learning tasks. Namely, given a source domain D_S and source learning task T_S, a target domain D_T and target learning task T_T, transfer learning aims to make a contribution (i.e., improvement) to the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S where $D_S \neq D_T$ or $T_S \neq T_T$ (Pan and Yang 2010).

Examples of transfer learning-based approaches that have been employed for hate speech detection include the Daume's Frustratingly Easy Domain Adaptation (FEDA), (Daume III, 2007), and word embeddings.

FEDA consists mainly of copying features several times to account for different domains, allowing the model to learn domain-dependent weights for each feature.

Karan and Šnajder (2018) applied FEDA framework to hate speech detection. Their method works by joining two data sets F_1 and F_2 from different domains in which their features are copied three times as follows: 1) unaltered instances from both domains; 2) F_1 specific features, which is 0 for all instances not from F_1; and 3) F_2 specific features, which is 0 for all instances not from F_2. Their study has tested the use of Support Vector Machines (SVM) for classification, concluding that domain adaptation boosts the classifier's performance significantly in six out of the tested nine cases (or data sets).

Sharifirad et al. (2018) have studied the effect of using word embeddings learned from data sets containing abusive language and compared it with lexicon-based features. Their experiments have shown that a Logistic Regressing classifier with abusive word embeddings trained with a couple of hundred training instances can outperform the same classifier with lexicon-based features on full data sets.

III. Negation Handling

Negated concepts and certainty modifiers are also encoded within the system, thus it enables them to make a distinction between negated/uncertain concepts and factual information which is crucial in information retrieval.

Identification of negation from open post documents is very challenging. Indeed, although one acknowledges the existence of several negation like connectives in natural language processing, e.g., no, not, none, xx-less, among others, many context related negated wordings are still difficult to recognize in line with Fregean injection "A negation may occur anywhere in a sentence without making the thought indubitably negative". Several text processing systems exploit hand-crafted rule-based negation/uncertainty detection modules. This includes, for instance, NegEx proposed by Chapman et al. (2001) that identifies negative scope.

Elkin et al. (2005) used a list of negation words and a list of negation scope-ending words to identify negated statements and their scope. This includes affixal negation constructs, e.g., either words with the prefixes un-, in-, dis-, a-, an- non-, im-, il-, ir-, or the suffix -less).

For example, unhappy can be paraphrased as not happy.

Huang and Lowe (2007) implemented a hybrid approach to automated negation detection by combining regular expression matching with grammatical parsing. In this respect, negations are classified on the basis of syntactic categories and they are located in parse trees. The method is shown to be able to identify negated concepts in radiology reports even when they are located at some distance from the negative term.

Moilanen and Pulman (2007) investigated the negation and its scope in the context of sentiment analysis. Danescu-Niculescu-Mizil et al. (2009) looked at the problem of finding downward-entailing operators that include a wider range of lexical items, including soft negators such as the adverbs "rarely" and "hardly". Wilson et al.(2005) suggested a supervised polarity classifier is trained with a set of negation features derived from a list of cue words and a small window around them in the text. On the other hand, WordNet (Miller, 1995) and thesauri such as Roget's already provide a collection of lexical negations. In WordNet, antonymy is defined as a lexical relation between individual lexemes that have clear opposite meanings (rather than between concepts, i.e. all the members of a synset). These 'direct antonym' pairs, such as wet:dry or long:short, are psychologically salient and have a strong associative bond between them resulting from their frequent co-occurrence (Fellbaum, 1998a).

On the other hand, 'Indirect antonyms' result from similarity relations defined for the members of these direct antonym pairs. For example, "moist" and "humid" are classified as semantically similar to wet, and are therefore indirect antonyms of the lexeme "dry".

IV. Data Set

In this paper, we consider two types of dataset: a publicly available dataset related to hate speech, and an artificially created dataset for unbalance class handling and hypothesis testing.

A. Wikipedia Comments Corpus

We have used annotated selected fragments of the Wikipedia comments corpus for personal attacks, aggression, and toxicity, available under free licenses on the Wikipedia Talk Corpus on Figshare[4].

Datasets related to Personal Attack, Aggression, and Toxicity contain 100k, 100k and 160k labeled comments from English Wikipedia, respectively. Each of this three dataset is

[4] https://figshare.com/projects/Wikipedia_Talk/16731

annotated by 10 annotators via Crowdflower on whether it contains the underlined property (Personal Attack, Aggression, Toxicity). Therefore, only comments that are supported by the majority of annotators (at least five annotators) are considered in our study.

B. *Artificial Dataset*

Ultimately, any classification-based approach for hate/abusive detection would require in addition to hate speech sample, non-hate speech as well. To build such a dataset, two approached have been adopted. The first one consists of introducing the negation in hate speech comment (personal attack, aggression or toxicity) dataset in a way to cancel out the hate-speech effect. The second one extracts random sentences from the generic Wikipedia comment dataset.

More specifically, for generating random sentence /post, we have used the random article generating link[5] that changes the article every reload of the page. Using this link, we have written a python code which generates random sentences /post from each article and saves the dataset into a csv file.

This results in new datasets of the csv form containing over 100k of well negated sentences of the three types of hate speech dataset.

An example of post negating is illustrated in Table 1.

Table1: Example of sentence and negation of sentence

Sentence	Negation
'This page will need disambiguation	This page will not need disambiguation
I removed from scratch. In addition to your reasons it just looks better without it.	I did not remove from scratch. In addition to your reason it just does not look better without it

One can observe the quality of negation in the second sentence, where the algorithm identified past sense verbal expression through part of speech tagging, resulting in adding 'did not' before the verbal expression, and then stemming the verb. The same applies to the present tense verb. This enables the output sentence to be well understood and one may not detect that this is a computer-generated negation.

A detailed description of the artificial negation generation is summarized in Algorithm 1.

The overall ideal consists of using sentence tokenizer and part-of-speech tree over each sentence. If the sentence does not contain a negation connective, then the approach would initially focus on adverb / adjective token of the sentence, if it exists, so that the adverb/adjective will be substituted by its antonym, if it exists, or preceded by a negation connective. In case there is no adverb/adjective in the sentence, then the negating will concentrate on introducing a negation connective on the verb token paying attention to the verb form.

Algorithm 1. Generic pseudo code for negation generation

Negating the hate datasets

i) Load a sentence and initiate a new list sentence negation.
ii) Perform Part-of-Speech Tree on the tokens of the sentence
iii) For each word in POS tags DO
  IF there is no negative connective in sentence.
    IF the word belongs to verb or adjective forms
    THEN
    EITHER Replace the verb / adjective by its Antonym, if it exists, OR Add a negation connective before it (paying attention to different forms of negations and stemming if it is a verb)
    END IF
  END
  ElseIf the word contains any negation connective
  THEN
  Remove the negation and restore word forms.
END

V. METHODOLOGY

A. *Preprocessing*

Both hate speech dataset and artificially generated dataset are preprocessed using standards NLP preprocessing tools involving mainly removal of unidentified characters, symbols, and words that do not have an entry in the dictionary. The latter consists of the WordNet lexical database and Wikipedia concept entries.

B. *Feature Engineering*

A set of features have been employed and evaluated for hate speech detection.

i) *TF-IDF features*.

We have used three types of these features: Word-level, N-Gram level (for N=2, 3), Character level. Word level TF-IDF feature assigns a score to every term in documents, while N-gram feature applies TF-IDF scoring to all 2-grams and 3-grams tokens extracted from the whole corpus dataset. Character Level TF-IDF provides a matrix representation of tf-idf scores of character level n-grams in the corpus. We restricted to 5000 features for each types to avoid the computational cost.

ii) Word Embeddings features

We have used the pre-trained word embedding FastText model (Bojanowski et al., 2016) (pre-trained on English Wikipedia articles), which makes use of character-level

[5] https://en.wikipedia.org/wiki/Special:Random

information. The key-advantage of such representation to other commonly employed embeddings (e,g., Word2vec, Glove) is the ability to handle words that do not have entry in the dictionary by exploring their n-gram structures.

iii) LIWC features

We have used empath-client tool[6]. The latter examines text through word categories, similarly to LIWC, and also engendering new word categories to use for a given inquiry. We have used both "positive emotions" and "negative emotions" categories because of its link to hate speech. For example, if a sentence analyzed by empath contains a higher number of negative-emotions compared to other sentences, this increases the chance that such sentence is a more hateful-sentence compared to other sentences.

C. Classification architecture and results

The overall architecture of the system is shown in Figure 1. Initially, we employed a random split of the original dataset into 70% for training and 30% for testing and validation ensuring the same proportion of dataset is taken from each category (Toxicity, Personal Attack, Aggression, and non-hate speech) to ensure balanced training. The developed approach makes use of two types of classification schemes. The first one is a binary classification, which distinguishes between hate and non-hate speech. In other words, all Personal-Attack, Aggression and Toxicity datasets are cast together into a single class pertaining to hate-speech. The second type distinguishes the three types of hate-speech and non-hate.

Two types of classifiers were implemented: Convolution Neural Network (CNN) and the recurrent neural network LSTM models. The CNN model uses the same architecture as in (Kim, 2014) where the input is represented as a concatenation of the words forming the post, except that each word is represented by its FastText embedding representation. A convolution operation with kernel size 3 was used together with a max-over-time pooling operation over the feature map. Dropout on the penultimate layer with a constraint on l2-norms of the weight vector was used regularization. Similarly, LSTM scheme is similar to that described in (Acken et al., 2018) with the difference in the word embedding representation as in our CNN model.

On the other hand, two baselines algorithms that use Linear Classifier (Logistic Regression) and Naives' Classifier were also considered for comparison purposes. The details of the implementation are reported in the Github page of this project.

The various features were examined by each classifier to test its accuracy and robustness.

The results of the binary classification hate versus non-hate speech identification (as well as Toxicity vs non-toxicity, aggression vs non-aggression, personal attack vs non-personal attack) are summarized in Table 2. We especially evaluate the performance in terms of F1-score and accuracy. Similarly, Table 3 summarizes the classifier performance in the case of multi-class classification through averaging all the three datasets.

At the same time, to test the performance of the constructed transfer learning scheme, we compared the result with that obtained when the training dataset is made of the total hate speech dataset and a random selection of the Wikipedia dataset of the same size. The negation based construction and non-negation based schemes are referred Dat1 and Dat0, respectively, in Table 3.

The performance of classifiers is provided in terms of F1-score and accuracy. F1-score corresponds to the harmonic mean of the precision and recall evaluation, namely:

$$F_1 = 2\frac{\mathrm{Pr}\,ecision.\mathrm{Re}\,call}{\mathrm{Pr}\,ecision + \mathrm{Re}\,call}$$

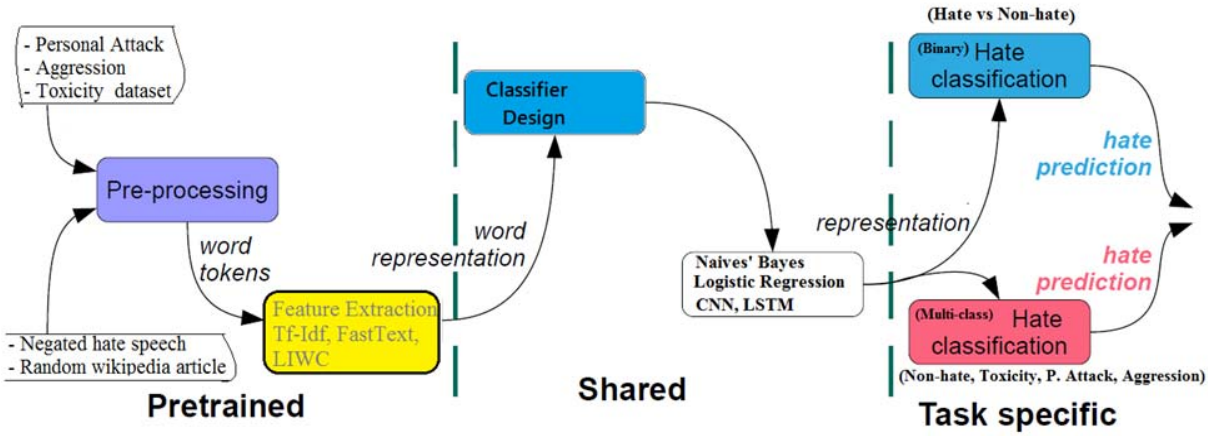

Figure 1. Conceptual approach of Transfer Learning scheme

[6] https://github.com/Ejhfast/empath-client

Table 2. Binary classification results

Features	CNN		LSTM		Log. Regr		Naives Bayes	
	F1 %	Acc %	F1 %	Acc %	F1 %	Acc %	F1 %	Ac %
	Personal – Attack							
Wo Tf-Idf					68	94	62	93
Ch Tf-Idf					68	93	57	91
N-gram					47	90	47	90
LIWC					72	92	71	93
FastText	70	88	71	88				
	Aggression							
Wo Tf-Idf					69	94	63	93
Ch Tf-Idf					69	94	58	92
N-gram					48	91	62	91
LIWC					70	92	70	93
FastText	72	91	70	90				
	Toxicity							
Wo Tf-Idf					69	94	64	94
Ch Tf-Idf					69	94	57	93
N-gram					48	91	47	91
LIWC					72	92	69	92
FastText	73	91	74	90				

Table 3. Overall multi-class classification

Classifier	F1-score		Accuracy	
	Dat1	Dat0	Dat1	Dat0
Naives'Bayes [Wo-Tf-Idf LIWC]	0.86	0.81	0.89	0.82
Logistic regression [Wo Tf-Idf LIWC]	0.87	0.82	0.90	0.81
CNN [Embeddings]	0.89	0.83	0.94	0.86
LSTM [Embeddings]	0.88	0.82	0.94	0.85

Discussion

The results highlighted in Table 1 and Table 2 indicate an increasing overall performance over the three classes (Toxicity, Aggression, and Personal Attack) for CNN and LSTM classifiers compared to baseline carried out by Naives' Bayes and Logistic Regression classifiers.

On the other hand, the analysis of the individual classifiers and impact of various features reveal the following. First, the accuracy performance of the classifiers shows a slight increase of baseline models (Naives' Bayes and Logistic Regression) over CNN and LSTM models, while the CNN and LSTM outperform the baseline models in terms of F1-score. Second, the use of Word-level Tf-Idf features in baseline models marginally outperforms that generated using character level Tf-Idf or N-gram (N=2,3) features. Third, the use of LIWC features in the baseline model induce increased performance in terms of F1-score evaluation. Fourth, although one can think of using the embedding features in baseline models, such an approach has not been conducted to ease comparison with other related works and avoid the computational explosion of the classifiers.

Fifth, the results highlighted in Table 3 have been obtained after testing several concatenations of the feature sets, and we only reported those features and results that yield the best overall classification results across the thee datasets. For instance, the best overall performance of Naives' Bayes classifier is obtained when using a concatenation of word'level Tf-Idf and LIWC as features for the classifier. Sixth, extra evaluation and testing are required to accurately quantify the effect of transfer learning scheme generated by a genuine change of initial dataset through negation handling to enforce a balanced training scheme for the classifiers.

VI. Conclusion

This paper deals with hate-speech online identification using original feature engineering and transfer learning scheme that makes use of negated dataset to enforce a balanced training paradigm. The methodology is tested on Wikipedia comment Corpus that involves three categories of hate speech: Toxicity, Personal Attack and Aggression. A convolution neural network together with long short term memory (LSTM) architectures that use FastText word embeddings features are contrasted with baseline algorithms constituted of Logistic Regression and Naives' Bayes classifiers. The feature sets constructed form Word-level Tf-Idf, Character level Tf-Idf and LIWC are compared and contrasted. The testing results demonstrate the feasibility of the developed transfer learning scheme, with negated dataset, to outperform standard random sampling data selection based approach. On the other hand, the superiority of the constructed CNN and LSTM based-classifiers in the overall classification of the three hate speech categories is clearly emphasized. This paves the way for future hybridization schemes that make use of more elaborated sentence constructs in the preparation of the testing dataset and feature selection process.

Acknowledgment

This work is partly supported by the EU Project YoungRes (#823701) on polarization detection.

References

Allan, J. (2013). The harm in hate speech. Constitutional Commentary, 29(1):59–80.

Alfred. V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling, volume 1*. Prentice-Hall, Englewood Cliffs, NJ.

Aken B. V. , J. Risch , R. Krestel , and A. Loser (2018), Challenges for Toxic Comment Classification: An In-Depth Error Analysis, Proceedings of the Second Workshop on Abusive Language Online (ALW2), pages 33–42 Brussels, Belgium

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114-133. https://doi.org/10.1145/322234.32224.

Bojanowski P., E. Grave, A. Joulin, and T. Mikolov (2016), Enriching word vectors with subword information. CoRRabs/1607.04606. http://arxiv.org/abs/1607.04606

Borschinger B. and Johnson M., (2011). A particle filter algorithm for Bayesian word segmentation. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, pages 10-18, Canberra, Australia.

Burnap P. and M. L. Williams. (2016), Us and them: identifying cyber hate on twitter across multiple protected characteristics. EPJ Data Science, 5(11):1–15, doi:10.1140/epjds/s13688-016-0072-6.

Burnap P. and Williams M. L., (2015). Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. Policy and Internet, 7(2):223–242.

Caruana R (1998) Multitask learning. Learning to learn, pp 95–133. Springer

Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan BB (2001), A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries. Journal of Biomedical Informatics, 34(5):301-310.

Chen Y., Y. Zhou, S. Zhu, and H. Xu (2012). Detecting offensive language in social media to protect adolescent online safety. In Proceedings of the 2012 ASE/IEEE International Conference on Social Computing and 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust, SOCIALCOM-PASSAT '12, Washington DC, pages 71–80.

Danescu-Niculescu-Mizil C., Lillian Lee, and Richard Ducott (2009), Without a 'doubt'? Unsupervised discovery of downward-entailing operators, In Proceedings of NAACL HLT 2009, arXiv:0906.2415v1 [cs.CL]

Djuric N., J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati (2015). Hate speech detection with comment embeddings. In Proceedings of the 24th International Conference on World Wide Web,WWW'15 Companion, pages 29–30, New York, NY, USA, 2015. ACM. doi:10.1145/2740908.2742760.

Elkin PL, Brown SH, Bauer BA, Husser CS, Carruth W, Bergstrom LR, Wahner-Roedler DL. (2005), A controlled trial of automated classification of negation from clinical notes. *BMC Med Inform Decis Mak* 2005, 5:13. doi:10.1186/1472-6947-5-13

Fellbaum, C. (1998), Computers and the Humanities 32: 209. https://doi.org/10.1023/A:1001181927857

Foong Y. J., and Oussalah M. (2017), Cyberbullying System detection and Analysis, in Proceedings of the IEEE European Intelligence and Security Informatics Conference, Athens, 10.1109/EISIC.2017.43

Harper M. (2014), Learning from 26 languages: Pro- gram management and science in the babel program. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, page 1. http://aclweb.org/anthology/C14-1001.

Galen A and Jianfeng G.. (2007). Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33-40.

Gambäck B., and Utpal Kumar Sikdar.(2017), Using convolutional neural networks to classify hate speech. In Proceedings of the First Workshop on Abusive Language Online, pages 85–90.

Goodman J., Andreas Vlachos, and Jason Naradowsky. 2016. Noise reduction and targeted exploration in imitation learning for abstract meaning representation parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, V1., p-1-11,* https://doi.org/10.18653/v1/P16-1001

Huang Y, Lowe HJ. (2007), A Novel Hybrid Approach to Automated Negation Detection in Clinical Radiology Reports. *Journal of the American Medical Informatics Association* 2007, 14(3):304-311.

Karan, M., and Šnajder, J. 2018. Cross-domain detection of abusive language online. In Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), 132–137.

Kim Y. (2014). Convolutional neural networks for sentence classification. In Proceedings, EMNLP.

Kwok I., and Yuzhou Wang (2013). Locate the hate: Detecting tweets against blacks. In Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, AAAI'13, Menlo Park, California, USA, pages 1621–1622.

Lenci A. (2018), Distributional models of word meaning, Annual Review of Linguistics,

Mahmud, A., K.Z. Ahmed, and M. Khan, M. (2008)."Detecting flames and insults in text". In: Proceedings of the6th International Conference on Natural Language Processing (ICON--2008), CDAC Pune, India, December 20 --22 (2008)

Manning C. D. and H. Shutze (1999), Foundation of Statistical Natural Language Processing, MIT Press.

Marneffe M., B. MacCartney, and C. D. Manning (2006), "Generating typed dependency parses from phrase structure parses," presented at the LREC, 449-454

Miller G. A. (1995), WordNet: a lexical database for English, Communications of the ACM, 38 (11), 39-41

Moilanen K. and Pulman, S. (2007), Sentiment composition. In: Proceedings of the 6th International Conference on Recent Advances in Natural Language Processing (RANLP). (2007) 378{382

Mutalik PG, Deshpande A, Nadkarni PM, (2001), Use of General-purpose Negation Detection to Augment Concept Indexing of Medical Documents: A Quantitative Study Using the UMLS. *Journal of the American Medical Informatics Association* 8(6):598-609.

Nobata C., Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. (2016) Abusive language detection in online user content. In Proceedings of the 25th International Conference on World Wide Web, WWW '16, pages 145–153, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee. do:10.1145/2872427.2883062

Nockleby, J. T. (2000), "Hate Speech" in Encyclopedia of the American Constitution, ed. Leonard W. Levy and Kenneth L. Karst, vol. 3. (2nd ed.), Detroit: Macmillan Reference US, pp. 1277–79.

Pan, S. J., and Yang, Q. (2010). A survey on transfer learning. Transaction on Knowledge and Data Engineering 22(10):1345–1359

Pennebaker, J.W., Boyd, R.L., Jordan, K., & Blackburn, K. (2015). *The development and psychometric properties of LIWC2015* . Austin, TX: University of Texas at Austin.

Rasooli M. S., and J. R. Tetreault.(2015). Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

Reynolds, K., Kontostathis, A., & Edwards, L. (2011). Using Machine Learning to Detect Cyberbullying. In Proceedings of the 2011 10th International Conference on Machine Learning and Applications and Workshops (pp. 241–244). Washington, DC, USA: IEEE Computer Society

Rie Kubota Ando and Tong Zhang. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817-1853.

Spertes, E. (1997). "Smokey: Automatic recognition of hostile messages". In Proceedings of the Eighth Annual Conference on Innovative Applications of Artificial Intelligence (IAAI), pp. 1058--1065

Salawu S, he Y, Lumsden J (2017). Approaches to Automated Detection of Cyberbullying: A Survey. IEEE Transactions on Affective Computing, 10.1109/TAFFC.2017.2761757

Sharon, Alina Dain (28 February 2013). "A Web of Hate: European, US, Laws Clash on Defining Policing Online Anti-Semitism", Algemeiner Journalm. http://www.algemeiner.com/2013/02/28/a-web-of-hate-european-u-s-laws-clash-on-defining-and-policing-online-anti-semitism/

Sharifirad, S.; Jafarpour, B.; and Matwin, S. (2018). Boosting text classification performance on sexist tweets by text augmentation and text generation using a combination of knowledge graphs. In Workshop on Abusive Language Online (ALW2), 107–114.

Warner, W., Hirschberg, J. (2012). Detecting hate speech on the world wide web, in: Proceedings of the Second Workshop on Language in Social Media, Association for Computational Linguistics. pp. 19–26.

Wilson, T., Wiebe, J., & Hoffmann, P. (2009). Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational linguistics, 35(3)*, 399-433.

Zhang Z, and Luo L. (2018), Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter, Semantic Web 1 (0) 1–5, arXiv:1803.03662v2