# Lexicon-Enhancement of Embedding-based Approaches Towards the Detection of Abusive Language

**Anna Koufakou, Jason Scott**
Florida Gulf Coast University
Fort Myers, Florida, USA
akoufakou@fgcu.edu

**Abstract**

Detecting abusive language is a significant research topic, which has received a lot of attention recently. Our work focuses on detecting personal attacks in online conversations. As previous research on this task has largely used deep learning based on embeddings, we explore the use of lexicons to enhance embedding-based methods in an effort to see how these methods apply in the particular task of detecting personal attacks. The methods implemented and experimented with in this paper are quite different from each other, not only in the type of lexicons they use (sentiment or semantic), but also in the way they use the knowledge from the lexicons, in order to construct or to change embeddings that are ultimately fed into the learning model. The sentiment lexicon approaches focus on integrating sentiment information (in the form of sentiment embeddings) into the learning model. The semantic lexicon approaches focus on transforming the original word embeddings so that they better represent relationships extracted from a semantic lexicon. Based on our experimental results, semantic lexicon methods are superior to the rest of the methods in this paper, with at least 4% macro-averaged F1 improvement over the baseline.

**Keywords:** Abusive Language Online, Personal Attacks, Embeddings, Lexicons

## 1. Introduction

The pervasiveness of social media and the increase in online interactions in recent years has also led to a surge of online abusive behavior, which can be exhibited in different forms: toxic comments, aggression, hate speech, trolling, cyberbullying, etc. Online abuse influences individuals and communities in many ways, from leading users to quit a particular online site, to move away from their home, or to even commit suicide. Governments as well as social media platforms are under pressure to detect and remove abusive posts and users. On the other hand, online communities thrive on free speech and would be damaged by flagging and removing innocent users. Many efforts have been made for these tasks, including automated systems as well as employing human moderators.

At a first glance, NLP models can learn linguistic patterns in conversations and detect offensive speech using features such as swear words or racial/sexist slurs. This becomes a difficult research problem as online conversational text contains casual language, abbreviations, misspellings, slang, etc. Additionally, there are gray areas which make it hard to determine if a comment is actually offensive or abusive.

Methods employing word or character embeddings have been used successfully in many NLP tasks such as sentiment analysis or classification. A great part of the current research in the field of abuse detection in online conversations is based on deep learning with embeddings; for example, see (Gamback and Sikdar, 2017; Pavlopoulos et al., 2017; Gunasekara and Nejadgholi, 2018; Mishra et al., 2018; Zhang et al., 2018) among others.

In this study, we explore different ways of using lexicons to enhance deep learning methods that use embeddings and how they apply to the task of detecting abusive language. Specifically, we apply Convolutional Neural Networks to automatically identify comments which contain personal attacks (Wulczyn et al., 2017). Our research follows two very different ways in the literature to employ lexicons. First, we look at the use of sentiment lexicons, a form of sentiment dictionary associating words with sentiments. We choose to follow the work by (Shin et al., 2017) which uses sentiment lexicon-based embeddings alongside word embeddings and integrates them in its convolutional model in different ways. Second, we explore semantic lexicons, which contain semantic relationships between words (for example, synonyms or antonyms). These methods essentially transform the word embeddings themselves so that they better reflect the semantic relationships of the words, based on the semantic lexicon. To the best of our knowledge, none of these ideas or the specific methods we use in this paper have been applied towards the detection of abusive language or related tasks. Our experiments show that the semantic lexicon based methods outperform the baseline CNN, while the sentiment lexicon methods perform the same or lower than the baseline. Additionally, the semantic lexicon methods offer an efficient and flexible approach to enhance embeddings (as also discussed in Vulić et al., 2018).

The following sections give an overview of related work, describe our corpus and the different approaches implemented and applied in this paper, and present our experimentation and results, followed by concluding remarks.

## 2. Related Work

Related work has focused on many tasks in the field of abuse detection, for example, detecting hate speech (e.g., Saleem et al., 2017), abuse (e.g., Waseem et al., 2017), gender- or ethnic-based abuse (e.g., Basile et al., 2019), and aggression (e.g., Kumar et al., 2018), among others.

There has been much work in literature with the Wikipedia Toxicity corpora used in our paper (see Section 3). The creators of these corpora, Wulczyn et al. (2017), explored character as well as n-gram based models with logistic regression and multi-layer perceptron models.

Gunasekara et al. (2018) used a related dataset from a Kaggle challenge[1] targeting a multi-label classification task. Some papers (e.g. Brassard-Gourdeau et al., 2019) focused on the Toxicity corpus, not the Personal Attacks corpus, which we use. Similarly to our work, Brassard-Gourdeau et al. (2019) utilized sentiment lexicons. They used the sum of the sentiment score of each word in the comment, which is quite different from the sentiment lexicon approaches we employed in this paper.

Recent research, such as (Pavlopoulos et al., 2017; Mishra et al., 2018; Kumar et al., 2019; Bodapati et al., 2019), included experimental results with the Personal Attacks corpus. Pavlopoulos et al. (2017) used a Recursive Neural Network (RNN) along with an attention mechanism. Mishra et al. (2018) built on the previous work by using character n-grams; their best algorithm achieved an F1 macro of 87.44 on the Personal Attacks data. Bodapati et al. (2019) compared different methods such as fasttext, CNN, and BERT using various combinations of word, character, and subword units and reported that they achieved state-of-the-art F1 macro (89.5) on the Personal Attacks data with BERT fine tuning. These papers either followed different preprocessing (for example, removed stop words or used bigrams) or a different experimentation setup (for example, artificially balanced the dataset or used a different split on the data), etc. Therefore, we cannot directly compare their results with ours. Ultimately, the goal of our paper is to explore the impact of using sentiment and semantic lexicons to enhance embedding-based methods, achieved by comparing these methods with our CNN baseline (see Section 4).

To the best of our knowledge, none of the sentiment or semantic lexicon ideas in this work have been applied towards abuse detection. Note that an early draft of this work with preliminary results was shown in (Koufakou and Scott, 2019). In the current paper, we present additional algorithms, extensive experimentation and results, and an in-depth examination of the results and our observations.

Beyond abusive language detection, one of the semantic lexicon approaches we used, retrofitting (Faruqui et al., 2015), has been successfully applied to the classification of pathology reports by (Alawad et al., 2018).

## 3. Corpus

For this paper, we focus on data released from the Wikipedia Detox Project[2] (Wulczyn et al., 2017). We obtain the data from figshare[3]. The three corpora included in the release are Personal Attacks, Aggression, and Toxicity; we focus on the Personal Attacks corpus. This contains more than a 100k comments from English Wikipedia labeled by approximately 10 annotators via Crowdflower on whether or not it contained a personal attack. The data also contains additional fields, such as the type of attack; we use only the comment text and whether it contained an attack or not (label).

First, we apply basic preprocessing to the comment text, for example: force lowercase, remove multiple periods or spaces, but keep the main punctuation. We do not remove stop words or fix spelling errors. We then extract single tokens (unigrams). Finally, we remove any records that ended up empty after the preprocessing. The resulting dataset contains a total of 115,841 text comments, each with annotations by about 10 human workers which indicated whether or not each worker believed the comment contained a personal attack. A comment in our data is labeled as an attack if at least 5 annotators labeled it as an attack. As a result, the dataset has the record and label characteristics shown in Table 1.

## 4. Approaches

In this section, we describe our baseline model, the sentiment lexicon approaches, and the semantic lexicon approaches. Figure 1 displays diagrams for the two different approaches explored in this paper.

### 4.1 Baseline

As our baseline, we employ a convolutional neural network (CNN) (Kim, 2014). This choice was made to follow (Shin et al., 2017) discussed in the next section. Additionally, in early experiments, our CNN did better on our data than other models we tried (e.g. RNN or GRU).

We first extract words from our corpus (as described in Section 3) and then create a word embedding matrix, which is the input to the model (see Section 5.1 for the embeddings we use in our experiments).

Word embeddings are first passed through an embedding layer, kept static in our experiments, before being fed as input into the convolutional layers. The window sizes of the convolutional filters are 3, 4, and 5: using multiple filters enables us to extract multiple features. We use Rectified Linear Unit (ReLU) as the activation function.

The feature maps generated by the convolutions are passed through a max pooling layer, which gives the maximum value from each feature map. The results are concatenated and passed to a soft-max fully connected layer to produce the classification.

### 4.2 Sentiment Lexicon Approaches

Sentiment lexicons generally associate each term in the lexicon with a positive or negative score. A term in the lexicon might be associated with a positive or negative label or it might be given an emotion (e.g. angry or happy) or it might have a continuous sentiment score.

For this section, we experiment with techniques from the paper by Shin et al. (2017). Figure 1(a) shows an overview of the sentiment lexicon approaches. These ideas involve creating sentiment embeddings from sentiment lexicons and then integrating the sentiment embeddings to the model (CNN) in different ways. For each word *w* in the corpus, we search for *w* in each sentiment lexicon; then, we construct a sentiment lexicon embedding by concatenating all the lexicon values corresponding to *w*.

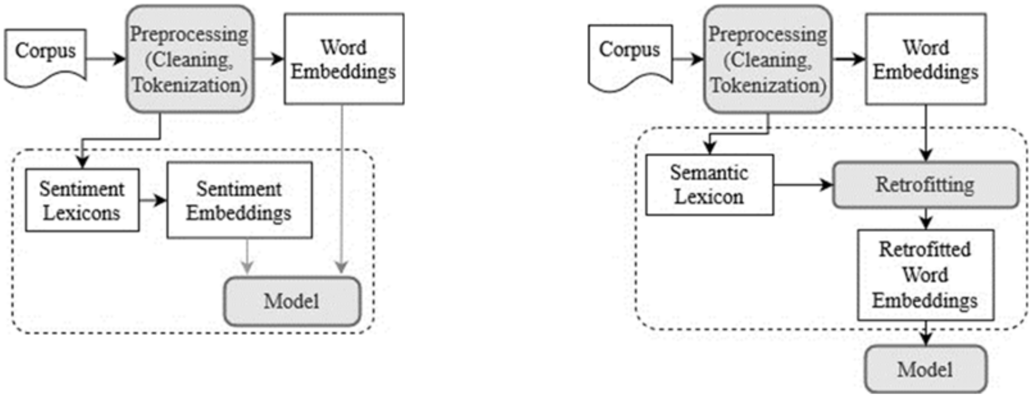
| | | |
|---|---|---|
| Attack | 14,205 | 12.3% |
| Not Attack | 101,636 | 87.7% |
| Total | 115,841 | 100.0% |

Table 1: The resulting Personal Attack dataset

[1] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

[2] https://meta.wikimedia.org/wiki/Research:Detox/Data_Release

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



(a) Sentiment Lexicon

(b) Semantic Lexicon

Figure 1. Block diagrams of the sentiment lexicon approaches versus the semantic lexicon approaches. The dashed-line rectangle indicates difference from the baseline. Grayed out lines for (a) indicate that the embeddings are used in different ways by the model (e.g., embeddings are concatenated or they pass through separate convolutions)

If *w* is not found in a lexicon, the value for that lexicon in the resulting embedding is 0. The lexicon embedding is a vector of dimensionality *l*, where *l* is the total number of sentiment lexicons. Finally, the word and sentiment lexicon embeddings are used by the model in different ways, described next.

The three approaches with sentiment lexicons based on (Shin et al., 2017) are briefly introduced below – the reader is referred to the original paper for more details:

**Naive Concatenation (NC):** This approach does not require any changes to the baseline model, as all of the modifications are on the embedding preparation stage. As described earlier, we extract sentiment lexicon entries for each word in our corpus. The entry from each lexicon is appended to the word embedding as an additional dimension before being fed into the embedding input layer. If *l* is the sentiment lexicon embedding dimensionality and *m* is the word embedding dimensionality, the resulting embedding for this approach is an (*l*+*m*)-dimensional combined embedding (word + sentiment).

**Separate Convolution (SC):** This approach does change the network from the baseline by adding a second input layer, and a second, parallel set of convolutional layers for the lexicon embeddings. The network has two inputs: one for the word embeddings and one for the lexicon embeddings, while the data input to each is, as before, the encoded text comments. The matrix of word embeddings and matrix of lexicon information each separately pass through convolutional layers, then are concatenated before continuing through the softmax layer of the network, as before.

**Embedding Attention Vector (EAV):** This approach utilizes the idea of attention. First, an attention matrix is constructed by performing multiple convolutions on the document matrix. Then, the attention vector is created by performing max pooling on each row of the attention matrix. The Embedding Attention Vector (EAV) is created by multiplying the transposed document matrix to the attention vector. EAVs are created for word and for lexicon embeddings. Finally, the resulting EAVs are appended to the penultimate layer of the network to serve as additional information for the softmax layer.

### 4.3 Semantic Lexicon Approaches

Semantic lexicons contain semantic relationships among the terms in the lexicon, for example synonyms. The main idea behind semantic lexicon-enhanced embeddings is that embeddings of words that are linked in the semantic lexicon should have similar vector representations (Faruqui et al., 2015).

The techniques presented in this section are quite different from the sentiment-lexicon approaches in the previous section: the techniques in this section use semantic knowledge to enhance (or transform) the word embeddings themselves rather than use the lexicon information in the learning process.

The block diagram in Figure 1(b) illustrates the semantic lexicon methods. The figure only refers to the first method in this section (retrofitting) for simplicity: any of the other methods can substitute it in the diagram. As shown in the diagram, the word embeddings pass through a retrofitting algorithm, resulting in the transformed embeddings (Retrofitted Word Embeddings) that are then fed into the model. These methods do not change the model itself, only the embeddings.

The three semantic lexicon approaches employed in this paper are briefly introduced below – the reader is referred to the original papers for more details:

**Retrofitting:** The first method in this section focuses on enhancing the word embeddings by "retrofitting" them to a semantic lexicon, as proposed by Faruqui et al. (2015). This method extracts synonym relationships from a semantic lexicon and "retrofits" the word embeddings based on belief propagation so that the vectors for synonym words are closer together in the vector space.

**ATTRACT-REPEL (AR):** While the Retrofitted embeddings focus on synonym relationships, more recent methods explore antonyms as well. The second method we explore is ATTRACT-REPEL (AR) proposed by Mrkšić et al. (2017). The key idea of this work is a process to fine tune pre-trained word embeddings also based on semantic constraints extracted from semantic lexicons. Given the initial vector space and collections of ATTRACT (synonym) and REPEL (antonym) constraints, the model gradually modifies the space to bring the designated word vectors closer together (synonyms) or further apart (antonyms).

**Post-Specialized:** Another issue for the semantic lexicon approaches is that semantic lexicons cover a small portion of the words in the corpus. This means that part of the word vectors resulting from retrofitting or AR (see above) are unchanged compared to the original word vectors, as a fraction of the words in the vocabulary are not found in the semantic lexicon.

This was addressed by the third method we explore, called Post-Specialized Word Embeddings, proposed by Vulić et al. (2018). This method extends the fine-tuning or specialization of embeddings to words not found in the external semantic lexicons. Essentially, it learns a mapping function based on the transformation of the "seen" words (e.g., the transformation from the original vectors into the AR vectors) and then applies this mapping to the vector space of the "unseen" words. The mapping is implemented as a deep feed-forward NN with non-linear activations.

# 5. Experiments

## 5.1 Experimental Setup

For our implementation, we use TensorFlow executed on Google Cloud TPUs on the TensorFlow Research Cloud[4], using a free trial of Cloud TPUs. We evaluate the network after 10,000 TPU steps of training with a randomly shuffled and batched training dataset, a learning rate of 0.001, dropout of 0.5, Adam optimizer, and 90-10 training-test split.

For the sentiment lexicon approaches (see section 4.2), we use the code provided online by Shin[5], though we had to make several modifications to adapt it to TPU-based code, handle old versions issues, etc.

For the semantic lexicon approaches (see section 4.3), we first construct our word embeddings as described in the next section. Then, we run the code provided by the authors of the corresponding papers[6] (with the parameters and lexical constraints/lexicons they provide) in order to "retrofit" or "specialize" our word embeddings as applicable. Finally, we use the resulting embeddings as input into the model.

## 5.2 Embeddings

We first pre-process the data, tokenize and generate word embeddings (see section 3 for our preprocessing and tokenization). Since the comments vary in length, we set the max document length to 400. Early on, we experimented with various types of embeddings (fasttext, pre-trained, etc.) and we saw that we obtain good results using gensim word2vec[7] on all tokenized sentences of our corpus (minimum word occurrences and iterations is set to 5). For all of our experiments, we use dimensionality of 200 or 300 (also used in the original papers) and report the best result.

| Lexicon | Type | Coverage |
|---|---|---|
| AFINN-96 | Sentiment | 3.3% |
| NRC | Sentiment | 11.1% |
| MSOL-June15-09 | Sentiment | 38.8% |
| Bing-Liu | Sentiment | 10.2% |
| PPDB-XL | Semantic | 67.5% |

Table 2: The coverage for the vocabulary in our corpus by each lexicon we use

Specifically for the Post-Specialized method (Vulić et al., 2018), we are unable to run the code using our own word embeddings (trained on our corpus, as described above), so we utilize the SGNS-BOW2 embeddings as provided with the post-specialization code[6] (Skip-Gram Negative Sampling, pre-trained on the Polyglot Wikipedia, 300-d). We see that this set of vectors covers about 90% of our vocabulary.

## 5.3 Lexicons

In this paper, we utilize the following sentiment lexicons for the sentiment lexicon methods (see section 4.2):

- **AFINN-96[8]:** The AFINN-96 sentiment lexicon (Nielsen, 2011) contains 3,382 words rated between -5 (most negative) and 5 (most positive).
- **NRC[9]:** The National Research Council Emotion Lexicon (Mohammad et al., 2013), commonly referred to as NRC EmoLex, contains 14,182 words labeled with eight emotions (anger, fear, etc.) and sentiment polarity (negative or positive).
- **MSOL-June15-09[10]:** The Macquarie Semantic Orientation Lexicon, or MSOL, contains a total of 76,400 entries either labeled as positive or negative (Mohammad, et al., 2009). It has 51,208 single-word entries.
- **Bing-Liu[11]**: The Bing-Liu Opinion contains 6,789 positive or negative words. The list was originally compiled as part of a study on mining and summarizing customer reviews but subsequently grew into a larger lexicon (Hu and Liu, 2004).

The sentiment lexicons above are preprocessed into lexicon embeddings using python code we wrote. Each lexicon is reduced to a key-value pairing of a word or phrase with its polarity value, which is -1 for negative polarity, 1 for positive polarity, or 0 for neutral. As described in section 4.2, every matching entry between our vocabulary and each sentiment lexicon is used to build the sentiment lexicon embeddings, following the work in the original paper by (Shin et al., 2017).

For retrofitting (Faruqui et al., 2015), we utilize the PPDB-XL[12] lexicon, as it was shown to have superior performance in the original paper and it had the best performance in our early trials. This lexicon is based on the paraphrase database (Ganitkevitch et al., 2013) with more than 220 million

[4] https://www.tensorflow.org/tfrc

[5] https://github.com/emorynlp/doc-classify

[6] https://github.com/mfaruqui/
https://github.com/nmrksic/attract-repel
https://github.com/cambridgeltl/post-specialisation

[7] https://radimrehurek.com/gensim/models/word2vec.html

[8] https://github.com/fnielsen/afinn

[9] http://saifmohammad.com/WebPages/lexicons.html

[10] https://www.saifmohammad.com/Lexicons

[11] http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar

[12] http://paraphrase.org/#/download

paraphrase pairs of English; of these, 8 million are lexical (single word to single word) paraphrases. For the rest of the semantic lexicon approaches (AR and Post-Specialized, see section 4.3), we use the lexical constraints as they are provided with the code of the respective paper.

We also provide the coverage of the vocabulary in our corpus by each lexicon we use (see Table 2). From the table, the coverage by the semantic lexicon is good, while for any sentiment lexicons, the word coverage is low. It is important to note that the sentiment lexicon percentages are similar to percentages in original paper for the related algorithms by (Shin et al., 2017).

## 5.4 Experimental Results and Discussion

Table 3 shows our results for the sentiment lexicon methods (see section 4.2) and the semantic lexicon methods (see section 4.3) versus our baseline (CNN, per the description in section 4.1).

We present results averaged over 10 different runs and reported accuracy, precision, recall, F1-score, and macro averaged F1-score (or F1-macro). As our dataset is very imbalanced (see Table 1), accuracy is not a good metric for comparison. The F1-score is the harmonic mean of the precision and recall. The macro averaged F1-score is the average of the F1-score for each class, averaged without taking class distribution into consideration. The macro-averaged F1 is better suited for showing the effectiveness of algorithms on smaller classes, which is important as we are interested in the small percentage of personal attacks in the data.

Overall, the sentiment lexicon techniques from (Shin et al., 2017) do not make a difference to the baseline or do worse than the baseline. For example, the baseline CNN with embeddings trained on our data has an F1-macro of 90.1 and all the sentiment lexicon methods have F1-macro from 85.7 to 90. Even through the low coverage of the words in our corpus by the sentiment lexicons (see Table 2) might seem like the likely reason for this, we note that the lexicon coverage in our paper is similar to the one reported in the original paper for these methods (Shin et al., 2017). One thing that we thought might improve the performance of these methods was to introduce more sentiment lexicons; however, we do not see a difference in performance from using one lexicon to using all four, so we do not further pursue this line of work (see section 0 for the sentiment lexicons we use and their coverage for our corpus). Extending our work to hate lexicons such as Hatebase[13] or HurtLex[14] is a line of future work.

On the other hand, all semantic lexicon approaches perform better than the baseline. The best performing semantic lexicon approach is the Post-Specialized Embeddings (Vulić et al., 2018) with a 95.1 F1-macro, followed closely by the other two semantic-based approaches (around 94 F1-macro) versus 90.1 for the baseline CNN with embeddings trained on our corpus. It is noteworthy that the Post-Specialized experiments in Table 3 use pre-trained embeddings (SGNS-BOW2), while the other two methods (Retrofitted and AR) use the respective techniques on the embeddings trained on our corpus (see section 5.2 for more information on the embeddings we used in our experiments).

A combination of the sentiment with the semantic lexicon approaches does not seem to yield better results: for example, applying first Naïve Concatenation (NC) of sentiment lexicon and word embeddings (see section 4.2) and then using the resulting embeddings in the Retrofitting approach (see section 4.3) shows no difference from the metrics shown in Table 3 for Retrofitted embeddings.

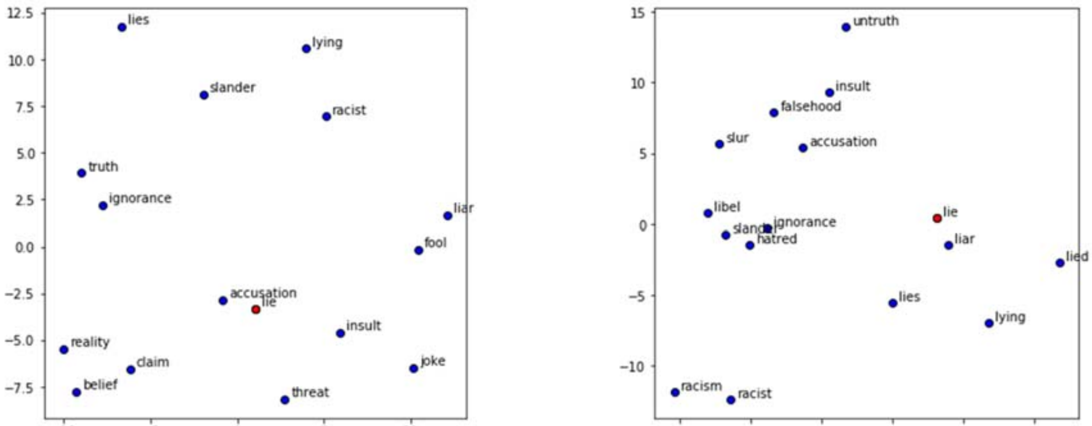
From the semantic lexicon approaches, it is noteworthy that the Retrofitting approach is the simplest of the semantic lexicon approaches, still it performs quite well (see Table 2). In order to explore the transformation of the words from our corpus in the vector space, we look at different word vectors before and after they are retrofitted to the semantic lexicon (Faruqui et al., 2015). All the results in the following discussion are according to cosine similarity.

| Approach | Embeddings | Model | Accuracy | Precision | Recall | F1 | F1-macro |
|---|---|---|---|---|---|---|---|
| Baseline | Word Embeddings | CNN | 95.9 ± 0.2 | 85.3 ± 0.8 | 80.1 ± 0.7 | 82.6 ± 0.6 | 90.1 ± 0.3 |
| Sentiment Lexicon | Sentiment<br>+<br>Word Embeddings<br>(Shin et al., 2017) | NC CNN | 95.9 ± 0.1 | 87.6 ± 0.8 | 76.8 ± 1.1 | 82.4 ± 0.5 | 90.0 ± 0.3 |
| | | SC CNN | 95.1 ± 0.1 | 85.1 ± 0.9 | 73.3 ± 1.4 | 78.7 ± 0.5 | 88.0 ± 0.3 |
| | | EAV CNN | 95.0 ± 0.1 | 83.9 ± 1.0 | 67.3 ± 1.6 | 75.5 ± 0.8 | 85.7 ± 0.4 |
| Semantic Lexicon | Retrofitted Word Embeddings (Faruqui et al., 2015) | CNN | 97.6 ± 0.1 | 93.8 ± 0.2 | 86.6 ± 1.0 | 90.0 ± 0.5 | 94.3 ± 0.3 |
| | Attract-Repel Word Embeddings (Mrkšić et al., 2017) | CNN | 97.4 ± 0.0 | 93.2 ± 0.4 | 85.9 ± 0.5 | 89.4 ± 0.1 | 94.0 ± 0.1 |
| | Post-Specialized (on SGNS-BOW2) Word Embeddings (Vulić et al., 2018) | CNN | **98.0** ± 0.0 | **95.3** ± 0.2 | **87.7** ± 0.7 | **91.4** ± 0.4 | **95.1** ± 0.2 |

Table 3: Results for our baseline, sentiment lexicon and semantic lexicon approaches (best results in **bold**)

[13] https://hatebase.org

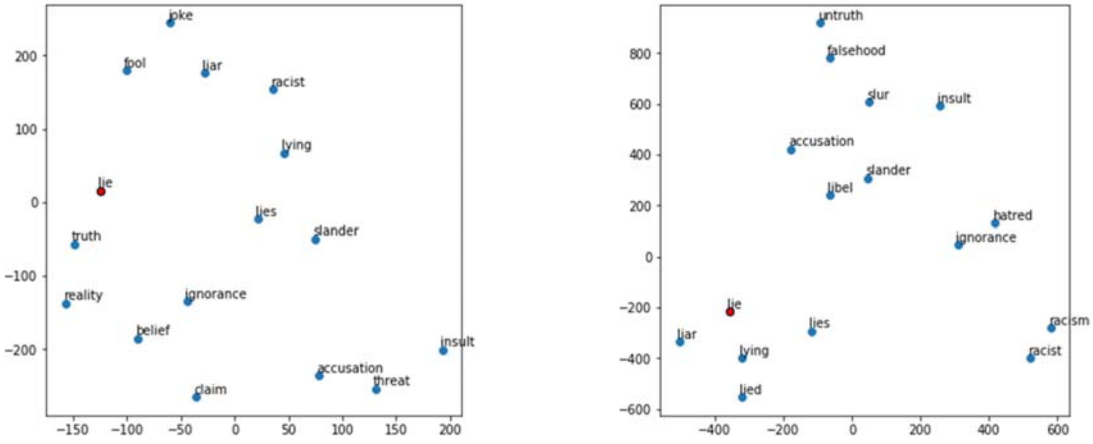
[14] https://github.com/valeriobasile/hurtlex



(a) Original

(b) Retrofitted

Figure 2. PCA projection of word embeddings (original vectors versus retrofitted vectors) for the fifteen closest words to the word ‘lie’ according to cosine similarity (300-d vectors)



(a) Original

(b) Retrofitted

Figure 3. t-SNE projection of word embeddings (original vectors versus retrofitted vectors) for the fifteen closest words to the word ‘lie’ according to cosine similarity (300-d vectors, t-SNE perplexity=5, iterations=1500)

The word ‘lie’ has the word ‘truth’ as its closest word in the original embeddings (similarity = 0.54), and the word ‘liar’ in the Retrofitted embeddings (similarity = 0.75). Also, the word ‘moron’ has the word ‘oxymoron’ as its closest word in the original embeddings (similarity = 0.73), and the word ‘retard’ in the Retrofitted embeddings (similarity = 0.87).

When we look at the twenty closest words of the word ‘moron’ using Retrofitted embeddings, the word ‘oxymoron’ is not in the list. When we pull the twenty closest words for the word ‘bye’, the results for the original embeddings include ‘wanker’, ‘sup’, ‘dickface’, and ‘slut’, while the results for the Retrofitted embeddings include no such words. Instead the Retrofitted results include ‘farewell’, ‘goodbye’, ‘ciao’ and ‘adios’, which are not in the original embedding results.

We additionally look at the same word-pairs with and without retrofitting. The similarity of ‘happy’ and ‘delighted’ is 0.54 in the original embeddings and 0.78 in the Retrofitted embeddings. The similarity of ‘moron’ and ‘idiot’ is 0.65 in the original embeddings and 0.84 in the Retrofitted embeddings.

At the same time, the similarity of ‘user’ and ‘admin’ is almost identical with and without Retrofitting (we checked and both words are in the semantic lexicon, PPDB-XL, used for the retrofitting). These results show that vectors for semantically related words do become more similar after retrofitting, while vectors for unrelated words stay unchanged.

Finally, we apply Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten and Hinton, 2008). Given the word ‘lie’ and its fifteen closest words (based on cosine similarity; fifteen was chosen for better visualization), Figure 2 shows a PCA projection and Figure 3 shows the t-SNE plot.

As shown in Figures 2 and 3, the fifteen closest words for the original embeddings contain close words such as ‘accusation, ‘insult, ‘joke’, ‘claim’. At the same time, the closest words for the Retrofitted embeddings are more similar to the word ‘lie’. In the t-SNE plot of the original embeddings (see Figure 3(a)), the work ‘lie’ is found close in the plot to ‘truth’, ‘reality’ or ‘fool’, while in the Retrofitted embeddings (see Figure 3(b)), it is close to ‘liar’, ‘lies’, and ‘lying’.

## 6. Conclusion

In this paper, we explore the use of lexicons, semantic or sentiment, for embedding-based methods towards the detection of personal attacks in online conversations (Wulczyn et al., 2017). The two types of approaches we employ are quite different in the type of lexicons they employ (sentiment or sematic) as well as how they use the lexicons in the learning process.

The sentiment lexicon approaches use the lexicons to create additional sentiment lexicon embeddings that are then used alongside the word embeddings in different ways (concatenation, separate convolutions or using attention mechanisms). The semantic lexicon methods use the original word embeddings and “enhance” them to better represent semantic relationships in the vector space, using the relationships extracted from the semantic lexicon.

Our experiments provide evidence that enhancing word embeddings using semantic lexicons shows promise for the task of abusive language detection. Besides improving detection accuracy for our data (in the form of F1-macro), these methods are fast and flexible, for example, they do not alter or depend on the type of learning model.

We plan to extend the approaches in this paper to enhance embeddings using hate speech lexicons, such as the ones presented in (Bassignana et al., 2018) and (Wiegand et al., 2018). We also plan to explore BERT fine tuning as in (Bodapati et al., 2019) and to explore the applicability of these methods in different data and languages other than English.

## 7. Acknowledgements

We gratefully acknowledge the Google Cloud Platform (GCP) research credits program and the TensorFlow Research Cloud (TFRC) program.

## 8. References

Alawad, M., Hasan, S. S., Christian, J. B., and Tourassi, G. (2018). Retrofitting word embeddings with the UMLS Metathesaurus for clinical information extraction. *Proceedings of the IEEE International Conference on Big Data (Big Data).*

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Pardo, F., and Sanguinetti, M. (2019). Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. *Proceedings of the 13th International Workshop on Semantic Evaluation SemEval-2019*.

Bassignana, E., Basile, V., and Patti, V. (2018). Hurtlex: A multilingual lexicon of words to hurt. *5th Italian Conference on Computational Linguistics, CLiC-it*.

Bodapati, S., Gella, S., Bhattacharjee, K., and Al-Onaizan, Y. (2019). Neural Word Decomposition Models for Abusive Language Detection. *Proceedings of the Third Workshop on Abusive Language Online, ALW3*.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5(135–146).

Brassard-Gourdeau, E., and Khoury, R. (2019). Subversive Toxicity Detection using Sentiment Information. *Proceedings of the Third Workshop on Abusive Language Online, ALW3*.

Faruqui, M., Dodge, J., Jauhar, J., Dyer, C., Hovy, E., and Smith, N. (2015). Retrofitting word vectors to semantic lexicons. *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics NAACL*.

Gamback, B., and Sikdar, U. (2017). Using convolutional neural networks to classify hate-speech. *Proceedings of the First Workshop on Abusive Language Online, ALW1*.

Ganitkevitch, J., Van Durme, B., and Callison-Burch, C. (2013). PPDB: The paraphrase database. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Glavaš, G., and Vulić, I. (2018). Explicit retrofitting of distributional word vectors. *Proceedings of the Annual Meeting of the Association for Computational Linguistics ACL*.

Gunasekara, I., and Nejadgholi, I. (2018). A Review of Standard Text Classification Practices for Multi-label Toxicity Identification of Online Content. *Proceedings of the 2nd Workshop on Abusive Language Online, ALW2*.

Hu, M., and Liu, B. (2004). Mining and summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*.

Kim, Y. (2014) Convolutional neural networks for sentence classification. *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*.

Koufakou, A., and Scott, J. (2019). Exploring the Use of Lexicons to aid Deep Learning towards the Detection of Abusive Language. *Proceedings of the 2019 Workshop on Widening NLP, ACL* (abstract).

Kumar, R., Ojha, A. K., Malmasi, S., and Zampieri, M. (2018). Benchmarking aggression identification in social media. *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying, TRAC-1.*

Mishra, P., Yannakoudakis, H., and Shutova, E. (2018). Neural Character-based Composition Models for Abuse Detection. *Proceedings of the Second Workshop on Abusive Language Online, ALW2*.

Mohammad, S., and Turney, P. (2013). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3), 436-465.

Mohammad, S., Dunne, C., and Dorr, B. (2009). Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*.

Mrkšić, N., Vulić, I., Séaghdha, D., Leviant, I., Reichart, R., Gašić, M., Korhonen, A., and Young, S. (2017). Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the Association for Computational Linguistics*, 5, 309-324.

Nielsen, F. (2011). A New ANEW: Evaluation of a Word List for Sentiment Analysis in Microblogs. *Proceedings* *of the ESWC2011 Workshop on "Making Sense of Microposts": Big Things Come in Small Packages*.

Pavlopoulos, J., Malakasiotis, P., and Androutsopoulos. I. (2017). Deep learning for user comment moderation. *Proceedings of the First Workshop on Abusive Language Online.*

Saleem, H. M., Dillon, K. P., Benesch, S., and Ruths, D. (2017). A web of hate: Tackling hateful speech in online social spaces. *arXiv preprint arXiv:1709.10159*

Shin, B., Lee, T., and Choi, J. (2017). Lexicon Integrated CNN Models with Attention for Sentiment Analysis. *Proceedings of the EMNLP Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*.

van der Maaten, L.J.P., and Hinton, G.E. (2008). Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research*. 9: 2579-2605.

Vulić, I., Glavaš, G., Mrkšić, N., and Korhonen, A. (2018). Post-Specialisation: Retrofitting Vectors of Words Unseen in Lexical Resources. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Waseem, Z., Davidson, T., Warmsley, D., and Weber, I. (2017). Understanding Abuse: A Typology of Abusive Language Detection Subtasks. *Proceedings of the First Workshop on Abusive Language Online ALW1*.

Wiegand, M., Ruppenhofer, J., Schmidt, A., and Greenberg, C. (2018). Inducing a lexicon of abusive words–a feature-based approach. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

Wulczyn, E., Thain, N., and Dixon, L. (2017). Ex-machina: Personal attacks seen at scale. *Proceedings of the 26th International Conference on World Wide Web WWW*.

Zhang, Z., Robinson, R., and Tepper, J. (2018). Detecting hate speech on twitter using a convolution-gru based deep neural network. *European Semantic Web Conference*.7