Pay "Attention" to Your Context when Classifying Abusive Language

Tuhin Chakrabarty[†] **Kilol Gupta**[†] **Smaranda Muresan**[†‡]
[†]Department of Computer Science, Columbia University
[‡] Data Science Institute, Columbia University
{tc2896, kilol.gupta, smara}@columbia.edu

Abstract

The goal of any social media platform is to facilitate healthy and meaningful interactions among its users. But more often than not, it has been found that it becomes an avenue for wanton attacks. We propose an experimental study that has three aims: 1) to provide us with a deeper understanding of current datasets that focus on different types of abusive language, which are sometimes overlapping (racism, sexism, hate speech, offensive language and personal attacks); 2) to investigate what type of attention mechanism (contextual vs. self-attention) is better for abusive language detection using deep learning architectures; and 3) to investigate whether stacked architectures provide an advantage over simple architectures for this task.

1 Introduction

Any social interaction whether in online forums, comment sections or micro-blogging platforms such as Twitter often involves an exchange of ideas or beliefs. Unfortunately, we often see that users resort to verbal abuse to win an argument or overshadow someone's opinion.

Natural Language Processing (NLP) could aid in the process of detecting and flagging abusive language and thus signaling abusive behaviour online. This is a particularly challenging task due to the noisiness of user-generated text and the diverse types of abusive language ranging from racism, sexism, and hate speech to harassment and personal attacks (Zeerak et al., 2017; Waseem and Hovy, 2016; Golbeck et al., 2017; Davidson et al., 2017; Djuric et al., 2015; Badjatiya et al., 2017; Park and Fung, 2017; Pavlopoulos et al., 2017). Zeerak et al. (2017) point out that different types of abusive language can be reduced to two primary factors:

1.	Obama is kinder to **islam** than any other future western leader is likely to be
2.	you can not even imagine how i think because i cannot imagine how anyone would take such a vile religion as **islam**

Table 1: Tweets where the word "islam" is used in two separate contexts: the top tweet is labeled as None while the bottom as Racism (Waseem and Hovy, 2016).

- Is the language directed towards a specific individual or entity or is it directed towards a generalized group?
- Is the abusive content explicit or implicit?

Table 1 shows two examples of tweets from the first large-scale Twitter abusive language detection dataset, where the second tweet expresses racism, while the first one does not (Waseem and Hovy, 2016). The usage of words in a particular context is important in determining the author's intended meaning. For example, the contexts of the word "islam" in the two tweets in Table 1 are different (a non-racist vs. a racist use of the word, respectively). Traditional bag-of-words models or simple deep learning models often cannot distinguish and handle such differences. This motivates us to explore deep learning models that use *contextual attention* for detecting abusive language and compare their performance against models with self-attention.

We make the following contributions:

- Conduct an empirical study to deepen our understanding of current datasets that focus on different types of abusive language, which are sometimes overlapping (racism, sexism, hate speech, offensive language and personal attacks). Show that our stacked Bidirectional Long Short Term Memory architecture with contextual attention is comparable to or out-

Proceedings of the Third Workshop on Abusive Language Online, pages 70–79
Florence, Italy, August 1, 2019. ©2019 Association for Computational Linguistics

performs state of the art approaches on all the existing datasets.

- Investigate what type of attention mechanism in deep learning architectures (contextual attention vs. self-attention) is better for abusive language detection. We show that contextual attention models outperform self-attention models on most cases (datasets and architectures), and present a thorough error analysis showing how contextual attention works better than self-attention particularly when it comes to modeling implicit abusive content.

- Investigate whether stacked architectures are better than simple architectures for abusive language detection when using Biderectional Long Short Term Memory (Bi-LSTM) networks. We show that stacked architectures are better than simple architectures on all datasets. In addition, we discuss the importance of pre-trained word embeddings for deep learning models. We make the code and all the experimental setups available in https://github.com/tuhinjubcse/ALW3-ACL2019.

2 Related Work

Work on abusive language detection has focused on specific types. Waseem and Hovy (2016) present a dataset of 16k tweets annotated as belonging to SEXISM, RACISM or NONE class and provide a feature engineered machine learning approach to classify tweets in the three classes. Davidson et al. (2017) uses a similar handcrafted feature engineered model to identify OFFENSIVE LANGUAGE and distinguish it from HATE SPEECH. Wulczyn et al. (2017) have contributed a Wikipedia Attacks dataset consisting of 115k English wiki talk page comments labeled as PERSONAL ATTACKS or NONE, while Golbeck et al. (2017) introduced a dataset labeled as HARASSMENT or NON-HARASSMENT. We present the first empirical investigation across all these existing datasets.

In recent years, deep learning models have been proposed for detecting different types of abusive language (Djuric et al., 2015; Badjatiya et al., 2017; Park and Fung, 2017). Djuric et al. (2015) propose an approach that learns low-dimensional, distributed representations of user comments in order to detect expressions of hate speech. Badjatiya et al. (2017) experiment with multiple deep learning architectures for the task of hate speech detection on Twitter using the same data set as Waseem and Hovy (2016) and report best F1-scores using Long Short Term Memory Networks (LSTM) and Gradient Boosting. Park and Fung (2017) use a Hybrid Convolution Neural Network (CNN) with the intuition that character level input would counter the purposely or mistakenly misspelled words and made-up vocabularies. Finally, Pavlopoulos et al. (2017) exploit deep learning methods with attention for abuse detection, where they use a self-attention model to detect abuse in news portals and Wikipedia. In this paper, we present an empirical study that investigates what type of attention mechanism (contextual vs. self-attention) is better for this task and whether stacked architectures are better than simple architectures. Yang et al. (2016) introduced a hierarchical *contextual attention* in a GRU architecture for document classification. The attention in this hierarchical model is both at the word and sentence level. For our study we use contextual attention only at word level because our Twitter datasets contains mostly single sentence tweets. Unlike Yang et al. (2016), we use a stacked Bidirectional Long-Short Term Memory (Bi-LSTM) network, and show that it is superior to using a single Bi-LSTM network.

3 Types of Abusive Language and Datasets

Abusive language can be of different types, and previous literature and datasets have focused on some of these types. Before introducing the existing datasets we use in our study, we provide the definitions for the types of abusive language used in existing work and examples for each type (Table 2):

- **Racism**: a belief that race is the primary determinant of human traits and capacities and that racial differences produce an inherent superiority of a particular race.

- **Sexism**: prejudice or discrimination based on sex; especially: discrimination against women.

- **Hate Speech**: is a language that is used to expresses hatred towards a targeted group or

Type	Example
Racism	The only reason the overall numbers increase is because Muslims breed like rats, just like their prophet told them to do. #Islam
Sexism	Don't ever let women drive, they'll break your arm!
Hate Speech	#westvirginia is full of white trash
Offensive Lang	I probably wouldnt mind school as much if we didnt have to deal with bitch ass teachers.
Harassment	yes ! whites who do not want to be a minority and will not accept being blended out of existence need to be shot ! #whitegenocide.
Personal Attack	what to do with elitist assholes who do not allow anybody else to edit certain pages? people such as alkivar? We must get rid of elitism, Wikipedia is a democracy for the contribution of ideas.

Table 2: Examples of different types of abusive language.

is intended to be derogatory, to humiliate, or to insult the members of the group (Davidson et al., 2017).

- **Offensive Language**: is a kind of abuse that causes someone to feel hurt, angry, or upset. It is usually rude or insulting and often very unpleasant.

- **Harassment**: is a type of abuse that is constructed with the identity of sincerely wishing to be part of the group in question, including professing, or conveying pseudosincere intentions, but its real intention(s) is/are to cause disruption and/or to trigger or exacerbate conflict for the purposes of amusement (Golbeck et al., 2017).

- **Personal Attack**: is a type of abuse that usually involves insulting or belittling one's opponent to invalidate his or her argument, but can also involve pointing out factual but ostensible character flaws or actions which are irrelevant to the opponent's argument.

We experiment with four benchmark datasets currently used in the related work on abusive language detection. Three of them are from Twitter (Table 3) and the fourth one from Wikipedia (Table 4), and together they showcase all the above mentioned types of abusive language.

- **D1** (Waseem and Hovy, 2016) — This is the first large-scale dataset for abusive tweet detection. Each of the $15,844$ tweets in the dataset is classified into three classes: RACISM, SEXISM, and NONE. Waseem and Hovy (2016) bootstrapped the corpus collection by performing an initial manual search of common slurs.

- **D2** (Davidson et al., 2017) — This dataset contains a total of $25,112$ tweets, each classified into one of the three classes: HATE SPEECH, OFFENSIVE LANGUAGE, and NEITHER. Davidson et al. (2017) began with a hate speech lexicon containing words and phrases identified by internet users as hate speech, compiled by Hatebase.org. They crawled 85.4 million using words from these lexicons before taking a random sample of 25k tweets manually coded by CrowdFlower (CF) workers.

- **D3** (Golbeck et al., 2017) — This dataset consists of $20,362$ tweets, with binary classes: HARASSMENT, and NON-HARASSMENT. Golbeck et al. (2017) (2017) settled on the following list of search terms ("#whitegenocide", "#fuckniggers", "#WhitePower", "#WhiteLivesMatter", "you fucking nigger", "fucking muslim", "fucking faggot", "religion of hate", "the jews", "feminist"). Though it produced a higher rate of tweets from alt-right / white nationalist tweeters, they were willing to accept a corpus that was not necessarily representative of all harassing content in order to achieve higher density.

- **D4** (Wulczyn et al., 2017) — The Wikipedia attacks dataset contains approximately 115K English Wikipedia talk page comments with binary classes: PERSONAL ATTACK, and NONE. Wulczyn et al. (2017) used a corpus that contains 63M comments from discussions relating to user pages and articles dating from 2004-2015. In order to get reliable estimates of whether a comment is a personal attack, each comment was labeled by at least 10 different Crowdflower annotators.

Table 3 shows the class-wise distribution for the three Twitter datasets **D1**, **D2** and **D3**, respectively. Table 4 refers to the class distribution of

	Class-wise Tweets			Total
D1	**Racism**	**Sexism**	**None**	15844
	1924	3058	10862	
D2	**Offensive**	**Hate**	**None**	25112
	19326	1428	4288	
D3	**Harass**		**N-harass**	20362
		5235	15127	

Table 3: Statistics of the Twitter datasets (**D1**, **D2**, **D3**).

D4	None	Personal Attack	Total
Train	61,447	8,079	69,526
Dev	20,405	2,755	23,160
Test	20,442	2,756	23,178

Table 4: Statistics of the Wikipedia dataset (**D4**).

Wikipedia comments labeled as PERSONAL ATTACKS or NONE (our **D4** dataset) divided among train, dev and test splits.

4 Methods

Long Short Term Memory Networks (LSTMs) (Hochreiter and Schmidhuber, 1997) are one of the most used deep learning architectures for different NLP tasks because of their ability to capture long-distance dependencies. For our task, we use Bidirectional LSTMs because of their inherent capability of capturing information both from the past and the future states.

Graves et al. (2013) show that LSTMs can benefit from stacking multiple recurrent hidden layers on top of each other. Thus, we choose to compare the simple Bi-LSTM architecture with a stacked Bi-LSTM architecture.

Attention mechanisms for deep learning models, including LSTMs serve two benefits: they often result in better performance in terms of metrics, and they provide insights into which words contribute to the classification decision which can be of value in applications and (error) analysis. There are several types of attention mechanisms. The key difference between *contextual attention* introduced by Yang et al. (2016) and self-attention is that it uses a word level context vector u_c that is randomly initialized and jointly learned during the training process (equation (2) vs. equation (3)).

$$u_i = tanh(W_h.h_i + b_h) \quad (1)$$

$$a_i^{contextual} = \frac{exp(u_i^T \mathbf{u_c})}{\sum_{j=1}^{T} exp(u_j^T \mathbf{u_c})} \quad (2)$$

$$a_i^{self} = \frac{exp(u_i^T)}{\sum_{j=1}^{T} exp(u_j^T)} \quad (3)$$

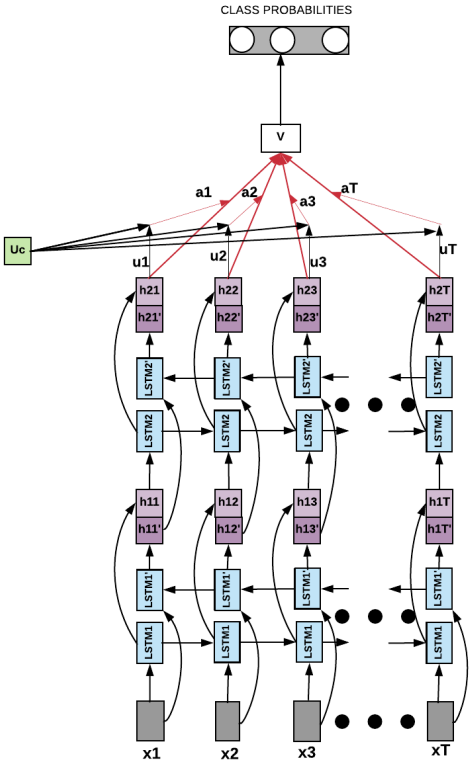

Figure 1: Architecture for Stacked BiLSTM + Word Level Contextual Attention. Figure is inspired by (Yang et al., 2016)

In this paper, we compare the effect of contextual attention as compared to self attention on both simple Bi-LSTMs and stacked Bi-LSTMs. Figure 1 shows the high-level architecture of our stacked Bi-LSTM model with contextual attention. The Bi-LSTM output h_i of each word x_i is fed through a Multi Layer Perceptron to get u_i as its hidden representation. u_c is our word level context vector that is a randomly initialized parameter of the neural network and is learned as we train our network. Once u_i is obtained we calculate the importance of the word as the similarity of u_i with u_c and get a normalized importance weight a_i through a softmax function. The context vector can be treated as a global importance measure of the words in the text. It takes into account which word to attend to based on how that word has been used in different contexts while training on the entire training set. The attention mechanism assigns a weight to each word annotation that is obtained from the Bi-LSTM layer. We compute the fixed representation $\mathbf{v}$ of the whole message as a weighted sum of all the word annotations, which is then fed to a final

fully-connected Softmax layer to obtain the class probabilities.

4.1 Implementation Details

We pre-process the text using `Ekphrasis` [1] — a text processing tool built specially for social media platforms such as Twitter.

For the Twitter datasets we experimented with word vectors that are initialized with pre-trained Twitter-specific embeddings (Baziotis et al., 2017), as well as ELMo embeddings (Peters et al., 2018), which are deep contextualized word representations modeling both complex characteristics of word use (e.g., syntax and semantics), and usage across various linguistic contexts.

For the Wikipedia Attacks dataset we relied on both fastText embeddings (Bojanowski et al., 2017) and ELMo embeddings. Out of vocabulary issues in pre-trained word embeddings are a major limitation for sentence representations. To solve this, we use fastText embeddings (Bojanowski et al., 2017), which rely on subword information. Also, these embeddings were trained on Wikipedia.

The embedding dimension of the words in our model for pre-trained Twitter embeddings and fastText embedding is set to 300, while for ELMo its set to 1024. We use a dropout rate of 0.25 and train the network using a learning rate of 0.001 for 10 epochs.

The results are reported by averaging over 10-fold cross-validation for datasets **D1** and **D3** and 5-fold cross-validation for **D2**. These protocols are consistent with all previously published results on the datasets. We report weighted-F1 scores for all the datasets to minimize the effect of class imbalance. For **D4** we train for 10 epochs and perform early stopping on a validation set. In order to be consistent with previous results we also report AUC scores for **D4** when comparing with state-of-the-art.

5 Results and Error Analysis

Our experimental study looks at several issues: the effect of contextual attention compared to self-attention; the stacked Bi-LSTM architecture compared to the simple Bi-LSTM architecture; the effect of pre-trained word embeddings; the effect of cross-datasets training/testing; and comparison of the best model against state-of-the-art results on each dataset.

[1] https://github.com/cbaziotis/ekphrasis

	Bi-LSTM + Self Attention	Bi-LSTM + Context Attention	Stacked Bi-LSTM + Self Attention	Stacked Bi-LSTM + Context Attention
D1	83.34	83.24	83.69	**84.25**
D2	89.27	89.83	89.95	**91.10**
D3	69.18	70.01	70.57	**72.75**
D4	94.22	94.87	95.11	**95.48**

Table 5: Weighted F1 scores on all datasets for all models.

Contextual vs. Self-Attention. Table 5 show all our models: simple Bi-LSTMs with self and contextual attention (column 2 and 3) as well as stacked Bi-LSTM models with self and contextual attention (column 4 and 5). We can see that contextual attention models outperform the self-attention models for both simple and stacked architectures on all datasets except on D1 for simple BiLSTM (i.e., columns 5 vs. 4, and 3 vs. 2; results are statistically significant with $p \leq .001$ using Chi Squared Test). For datset D1 and D2, we have several classes of abusive language (RACISM, SEXISM for **D1**; and HATE SPEECH and OFFENSIVE LANGUAGE in **D2**). Thus, we wanted to see the performance of the contextual vs self-attention on these finer grained classes (Tables 7 and 8). Table 7 shows that the contextual attention models have significant improvement over the models with self-attention when it comes to identifying RACISM and SEXISM. For the **D2** dataset we see that the most affected class is HATE SPEECH, the primary reason for this being that the percentage of data labeled as hate-speech is really small (5.6%). Even then the contextual attention models perform better than the ones using self-attention as shown in Table 8.

One of the main questions is **Why contextual attention is better than self attention?** What is there in the structure of context attention that leads to performance improvements over self attention? As discussed in Section 4, the context vector can be treated as global importance measure of words in text because it takes into account which word to attend to based on how that word has been used in different contexts while training on the entire training set. To highlight this behavior, in Table 6 we show several tweets from our data sets along with their true label. These tweets were predicted correctly by the context attention but incorrectly

	TWEET	TRUE LABEL	POTENTIAL EXPLANATION FOR PREDICTION
D1	Turkey and Egypt used to be mostly Christian and the muslims have mostly exterminated them	Racism	There are no jews in Saudi or many of the gulf states because the muslims exterminated them
			Jews used to live on 40% of the Arabian peninsula. muslims have virtually exterminated them
D1	Science was moving forward in India and Persia before islam , islam only slowed it down	Racism	People were making scientific discoveries , including algebra , before islam
			And notice that the Persian culture was more advanced and advancing and discovering before islam
D1	I don't think women can make tough military decisions. notice hilary's face during the bin laden raid	Sexism	i am not trying to be sexist but i do not think women should announce football games
			call me sexist but i do not think women should be allowed to grow beards
D2	Sonnen is a faggot	HateSpeech	Kanye West is a faggot
			Joshua is a faggot . just suspend him on those grounds

Table 6: Examples correctly classified by : Context Attention (CA) but mis-classified by Self Attention (SA)

	RACISM	SEXISM
Stacked Bi-LSTM + Context Attention	**79**	75
Stacked Bi-LSTM + Self Attention	74	73
Single Bi-LSTM + Context Attention	76	75
Single Bi-LSTM + Self Attention	73	73

Table 7: F1 scores of RACISM and SEXISM on **D1** on one of the test splits

	HS	OL	NONE
Stacked Bi-LSTM + Context Attention	**40**	95	90
Stacked Bi-LSTM + Self Attention	35	95	88
Single Bi-LSTM + Context Attention	38	95	88
Single Bi-LSTM + Self Attention	34	95	86

Table 8: F1 scores of OFFENSIVE LANGUAGE (OL) and HATE SPEECH (HS) and NONE on **D2** on one of the test splits.

by Self attention. The first three tweets were predicted as NONE by the self attention model while the last tweet was labeled as OFFENSIVE LANGUAGE. The "potential explanation for prediction" column shows tweets from the training data that have the same gold label and that are similar to the tweets in the test set shown in column 2, suggesting that the context attention indeed encapsulates the information by looking at examples globally through the training data, unlike self attention which only focuses on words for that particular tweet while trying to classify it.

DataSet	ELMo (Wiki)	Glove_Twitter
D1	83.10	**84.25**
D2	88.44	**91.10**
D3	68.78	**72.75**

Table 9: Weighted F1 scores comparing pre-trained embeddings on the Twitter datasets.

Stacked vs Simple Bi-LSTM. Table 5 shows that the stacked Bi-LSTM models outperformed the simple Bi-LSTM models, when using the same type of attention mechanism on all datasets (columns 5 vs. 3 and 4 vs. 2; results are statistically significant, with $p \leq .001$ using Chi Squared Test). When looking at Table 7 and 8, we notice that the stacked Bi-LSTM models do better than the simple Bi-LSTMs when using the same type of attention, only for the RACISM class and the HATE SPEECH class. The best performing model is the stacked Bi-LSTM with contextual attention.

Effect of pre-trained embeddings. The models presented above in Table 5 used Twitter-specific pre-trained embeddings for datasets **D1**, **D2** and **D3** and fasText embeddings trained on Wikipedia for **D4** (i.e., pre-trained embeddings from the same genre as the datasets). To compare the effect of pre-trained embeddings, we chose to compare our best model (Stacked Bi-LSTM with contextual attention) with the same model but trained using ELMo embeddings on the Twitter datasets. ELMo embeddings have been shown to outperform other types of embeddings on a variety of NLP tasks (Peters et al., 2018). The currently released ELMo embeddings are trained on news crawl data and Wikipedia and not on Twitter, which allows us to test the effect of pre-trained embeddings (genre,

Training Dataset	Weighted F1
D1+D2	64.50
D3	**72.75**

Table 10: Cross-datasets training (same CV test splits of D3)

method of training) on the performance of the deep network architectures. Table 9 shows that using the ELMo pre-trained embeddings instead of Twitter pre-trained embeddings lead to a statistically significant decrease in performance on all the Twitter datasets, with the biggest drop on **D2** and **D3**, which are the datasets on hate speech and harassment.

Cross datasets training/testing. The definition of the category HARASSMENT in the **D3** dataset states that it refers to language that is deeply racist, misogynistic or homophobic, bigoted, involved threats or hate speech. Given that the datasets D1 and D2 contain the categories RACISM, SEXISM and HATE SPEECH and are also from Twitter, we wanted to conduct a study where we train on **D1** and **D2** and test on **D3**. We considered data labelled as RACISM, SEXISM and HATE SPEECH as HARASSMENT and NONE as NON-HARASSMENT. This led to consistent class balance across train and test. The cross validation setting used for individual experiments on D3 was maintained here as well. Table 10 demonstrates that cross dataset training leads to worse performance when it comes to abusive language detection, showing that each dataset has its own particularities on defining and collecting the data.

Comparison with State-of-the-Art. We compare our best model (stacked Bi-LSTM with contextual attention) with various state-of-the-art models developed for each of the datasets we considered. For the Twitter datasets we compared against (1) an n-gram model with various linguistic features (Waseem and Hovy, 2016), (2) another model with hand-crafted features including n-grams, POS tags (Davidson et al., 2017); (3) a hybrid CNN model (Park and Fung, 2017), and (4) an LSTM model with an additional classifier using Gradient Boosting trees with LSTM embeddings as features (Badjatiya et al., 2017). Table 11 shows the weighted-F1 obtained by the models on the three Twitter datasets (**D1**, **D2**, **D3**). Note that none of the existing approaches show results on all the datasets. Thus, we report results using their

	D1	**D2**	**D3**
Majority Baseline	56.0	66.0	63.0
(Waseem and Hovy, 2016)	73.8†	82.3	63.0
(Davidson et al., 2017)	78.0	90.0†	63.8
(Park and Fung, 2017)	82.7†	88.0	68.6
(Badjatiya et al., 2017)	**93.1**†	NA	NA
(Badjatiya et al., 2017)_OurRep	81	88.0	67.4
Our Model	84.2	**91.1**	**72.7**

Table 11: Comparison of our best model with state-of-the-art models on the three Twitter datasets. †Results as reported in the respective papers.

METHOD	DEV	TEST
Majority Baseline	51.23	50.40
Our best model	97.39	97.44
(Wulczyn et al., 2017)	96.59	96.71
(Pavlopoulos et al., 2017)	**97.46**	**97.68**

Table 12: Comparisons with state-of-the-art models on **D4** DEV and TEST.

publicly available implementations on the remaining datasets, and highlight for which datasets they report results in their work.

Most abusive language datasets are highly imbalanced and thus we also report the scores for the majority baseline in Table 11 and Table 12. For D1, D3, D4 we predict everything as the majority class (Non-Abusive) and for D2 everything as offensive language. We see our best model beats the majority baseline by a huge margin. Our model obtains significantly better results ($p \leq .001$ using Chi Squared Test) than all the existing models on the datasets **D2** and **D3**. Notably, the improvements over the previous best performing models on these datasets are 1 F1 point and 2 F1 points respectively. On dataset **D1**, our model is outperformed by (Badjatiya et al., 2017), who mentioned that using Gradient Boosting Trees with LSTM embeddings boosted their model's performance by 12 points in weighted-F1. Unfortunately, while trying to replicate their results on dataset **D1**, we found no improvement from their simple LSTM model (the authors did not released the Gradient Boosting Trees with LSTM embeddigs implementation so we reimplemented that ourselves; weighted-F1 score of 81). Thus, for this model where we could not replicate the results on the original dataset, we report both the original results on that dataset and our re-implementation results on all datasets.

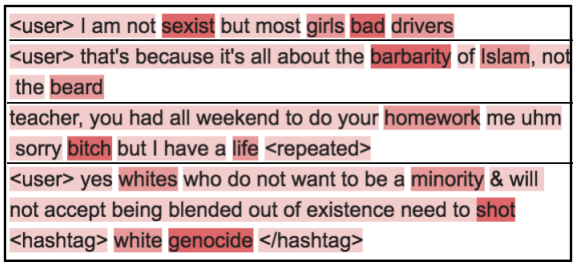

Figure 2: Attention heat map visualization demonstrating the focus on abusive-language signaling words in various tweets.

For the **D4** dataset which is Wikipedia, we compared our best model (stacked Bi-LSTM with contextual attention) with the existing models on this dataset. Wulczyn et al. (2017) use a Multilayer Perceptron over char n-grams as features and reported results only on the Dev set. We use their online implementation to report results on the test set. Pavlopoulos et al. (2017) use a deeper self attention mechanism and report results both on the Dev and Test sets. Both approaches report results using AUC. Table 12 shows that our model outperforms (Wulczyn et al., 2017) and is comparable to (Pavlopoulos et al., 2017).

6 Visualizing the Contextual Attention Weights

The contextual attention mechanism enables our model to focus on the relevant parts of the text (e.g., tweet) while performing the prediction task. As shown in Figure 2 and 3 our model learns to focus on relevant keywords that govern the abusive nature of a text. The color intensity here denotes the relative weight assigned to words. In figure 2, we see four tweets where the first tweet is labeled as SEXISM and the second tweet is labeled as RACISM from the **D1** dataset (Waseem and Hovy, 2016). The third tweet is a tweet from the **D2** dataset (Davidson et al., 2017) labeled as OFFENSIVE LANGUAGE and correctly identified by our model. The last tweet is from the **D3** dataset (Golbeck et al., 2017) labeled as HARASSMENT and correctly identified by our model. Figure 3 shows two such comments from the Wikipedia attacks dataset (**D4**), which were classified correctly by our model.

Moreover, it it encouraging to see that the contextual attention assigns higher weight to potentially abusive words when used with an abusive meaning. For example, refer to the two tweets

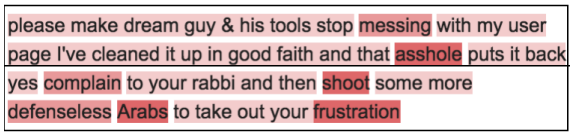

Figure 3: Attention heat map visualization demonstrating the focus on abusive words in Wikipedia Personal Attacks dataset.

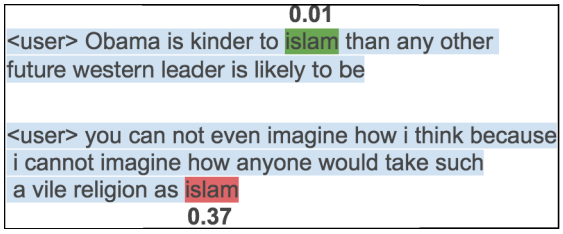

Figure 4: Attention weights learned by our model for the same word "islam" on two tweets.

in figure 4. The first tweet belongs to the NONE class while the second tweet belongs to RACISM class. The word "islam" may appear in the realm of racism as well as in any normal conversation. We find that our model successfully identifies the two distinct contextual usages of the word "islam" in the two tweets, as demonstrated by a much higher attention weight in the second case and a relatively smaller one in the first case.

7 Conclusion

Abusive language detection on the web is challenging for two reasons: (1) the inherent nature of noise in online discussions and (2) the contextual use of words that convey abuse only in certain contexts. We presented an extensive empirical study on several existing datasets that reflect different but possibly overlapping types of abusive language. We show that contextual attention is better than self-attention for deep learning models and using a stacked architecture outperforms a simple architecture (our basic architecture being a Bi-LSTM). We also show that using pre-trained embeddings from the same genre as the datasets is more important than better models for training the embeddings. Our best performing model, the stacked Bi-LSTM model with contextual attention is comparable to or outperforms state-of-the-art models on all the datasets. We also conduct a cross-dataset training/testing experiment that highlights the particularities of various datasets when it comes to the collection and labeling of abusive language. We present an error

analysis of the results and a visualization of the contextual attention weights — an important step towards better interpretation of any deep learning models.

While we notice that the visualization of attention weights is indicative of the classifier decision for multiple examples based on our context-attention model, some recent work has claimed that attention is not explanation (Jain and Wallace, 2019). As a future step, we would like to conduct experiments to measure the correlation between the highest attention weights chosen by models and humans (Ghosh et al., 2017) to further strengthen the interpretability of the attention-based models.

References

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760. International World Wide Web Conferences Steering Committee.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *(ICWSM 2017)*, pages 3952–3958. Proceedings of the Eleventh International AAAI Conference on Web and Social Media.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web*, pages 29–30. ACM.

Debanjan Ghosh, Alexander Richard Fabbri, and Smaranda Muresan. 2017. The role of conversation context for sarcasm detection in online interactions. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 186–196, Saarbrücken, Germany. Association for Computational Linguistics.

Jennifer Golbeck, Zahra Ashktorab, Rashad O Banjo, Alexandra Berlinger, Siddharth Bhagwan, Cody Buntain, Alicia A Cheakalos, Paul Geller, Quint Gergory, Rajesh Kumar Gnanasekaran, Raja Rajan Gunasekaran, Kelly M Hoffman, Jenny Hottle, Vichita Jienjitlert, Shivika Khare, Ryan Lau, Marianna Martindale, J Shalmali Naik, Heather L Nixon, Piyush Ramachandran, Kristine M Rogers, Lisa Rogers, Meghna Sardana Sarin, Jayanee Shahane, Gaurav Thanki, Priyanka Vengataraman, Zijian Wan, and Derek Michael Wu. 2017. A large human-labeled corpus for online harassment research. In *ACM*, pages 229–233. Proceedings of the 2017 ACM on Web Science Conference.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. *International Conference on Acoustics, Speech, and Signal Processing*.

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. In *Neural computation, 791*, pages 1735–1780.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *https://arxiv.org/abs/1902.10186*.

Ji Ho Park and Pascale Fung. 2017. One-step and two-step classification for abusive language detection on twitter.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1391–1399. International World Wide Web Conferences Steering Committee.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489. Association for Computational Linguistics.

Waseem Zeerak, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse:a typology of abusive language detection subtasks. In

Proceedings of the First Workshop on Abusive Language Online, pages 78–84.