Self-Contextualized Attention for Abusive Language Identification

Horacio Jarquín-Vásquez and **Hugo Jair Escalante** and **Manuel Montes-y-Gómez**
Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE)
Luis Enrique Erro #1, Sta María Tonanzintla, 72840 San Andrés Cholula, Pue.
{horacio.jarquin, hugojair, mmontesg}@inaoep.mx

Abstract

The use of attention mechanisms in deep learning approaches has become popular in natural language processing due to its outstanding performance. The use of these mechanisms allows one managing the importance of the elements of a sequence in accordance to their context, however, this importance has been observed independently between the pairs of elements of a sequence (self-attention) and between the application domain of a sequence (contextual attention), leading to the loss of relevant information and limiting the representation of the sequences. To tackle these particular issues we propose the self-contextualized attention mechanism, which trades off the previous limitations, by considering the internal and contextual relationships between the elements of a sequence. The proposed mechanism was evaluated in four standard collections for the abusive language identification task achieving encouraging results. It outperformed the current attention mechanisms and showed a competitive performance with respect to state-of-the-art approaches.

1 Introduction

The integration of social media platforms into the everyday lives of billions of users has increased the number of online social interactions, promoting the exchange of different opinions and points of view that would otherwise be ignored by traditional media. The use of these social media platforms has revolutionized the way people communicate and share information. Unfortunately, not all of these interactions are constructive, as the presence of Abusive Language (AL) has spread to these media.

AL is characterized by the presence of insults, teasing, criticism and intimidation (Cecillon et al., 2019). Mainly, it includes epithets directed at an individual's characteristic, which are personally offensive, degrading and insulting. Because of its negative social impact (Kumar et al., 2018), the automatic identification of AL has stimulated the interest of social media companies and governments (Hinduja and Patchin, 2010). Derived from this, multiple efforts have been made to combat the proliferation of AL, starting from the codes of conduct, norms and regulations in the content publication on social media[1], to the use of Natural Language Processing (NLP) for the computational analysis of language (Schmidt and Wiegand, 2017).

Concerning the several efforts and approximations made by the NLP community, one of the most relevant issues in the AL identification task is to distinguish between the use of profane words and vulgarities in offensive and non-offensive texts. This indicates that the importance and interpretation of each word is highly context dependent, and, accordingly, this particular issue evidences one of the reasons why traditional bag-of-words methods tend to generate many false positives in their predictions. Few works related to this task have explored the importance of words according to their context; particularly, the use of Deep Learning (DL) approaches with the addition of the Attention Mechanism (AM) has been explored as an alternative to solve this issue (Pavlopoulos et al., 2017; Chakrabarty et al., 2019; Jarquín-Vásquez et al., 2020).

The idea behind the use of the AM is to provide the classification model with the ability to focus on a subset of inputs (or features), handling in this way the importance of words in accordance to their context. Due to their outstanding performance in many NLP tasks, several AM have been proposed in recent years (Chaudhari et al., 2020), which can be divided into two main approaches: Self-Attention (SA) (Vaswani et al., 2017) and Contextual Attention (CA) (Yang et al., 2016) mechanisms. Specifically, SA takes the relationships among words within the same sentence, whereas,

[1] http://ec.europa.eu/justice/fundamental-rights/files/hate_speech_code_of_conduct_en.pdf

Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media, pages 103–112
Online Workshop, June 10, 2021.

CA selectively focuses on words with respect to some external query vector, which adjusts according to the training task. The more important the word is in determining the answer to that query, the more focus it is given.

Despite their outstanding performance, both approaches have their own limitations. On one hand, CA ignores the internal relationships between the words of a sequence, correspondingly, SA does not consider the global relationships within the words of different sequences, which causes the loss of relevant information in the application domain (training task). Clearly, the limitations of these AM are complimentary and a hybrid AM could overcome the individual issues. In this work we extend the use of the AM by proposing the Self-Contextualized Attention (SCA) mechanism, an AM that trades off the previous limitations, by taking advantage of both SA and CA mechanisms. The proposed SCA mechanism is designed to be applied to any sequence of word encoding features, nevertheless, due to the high context-dependency of words that this specific task has, in this work we exclusively focus on the AL identification task.

The main contributions in this paper are: After identifying a Deep Neural Network (DNN) architecture that is rather stable and well-performing, we propose and integrate the SCA mechanism into the DL architecture, subsequently we conduct a quantitative and qualitative study of the effectiveness of our proposed AM against the use of SA, CA and some other novel approaches to the AL identification task. To the best of our knowledge this is the first effort in combining both AM variants.

This paper is organized as follows: In Section 2, we present some previous works related to the AL identification task, along with other hybrid AM approaches. In Section 3, we describe our proposed SCA mechanism, as well as the employed classification framework; in Section 4, we present the datasets used to evaluate our SCA mechanism, their implementation details, as well as the external resources fed to the classification framework. Section 5 reports and discusses our quantitative and qualitative results. Finally, Section 6 summarizes our findings and discusses future work.

2 Related work

Considering the well-acknowledged increase of AL on social media platforms, several datasets (Zeerak and Dirk, 2016; Davidson et al., 2017; Marcos et al., 2019) and evaluation campaigns (Fersini et al., 2018; Kumar et al., 2018; Aragón et al., 2020), have been proposed in order to mitigate the impact of such a kind of messages.

The detection of AL has been mainly addressed from a supervised perspective, considering a great variety of features. Initial works used a combination of hand-crafted features such as bag-of-words representations, considering word and character n-grams (Burnap and Williams, 2016), as well as, syntactical and linguistical features (Nobata et al., 2016). Aiming to improve the generalization of the classifiers, some other works have explored the use of DL by taking word or character sequences from texts to learn abusive patterns without the need for explicit feature engineering; the use of word embeddings as features predominates in these works (Zhang et al., 2018; Saksesi et al., 2018; Amrutha and Bindu, 2019). More recently, there has been a trend within the NLP community regarding the use of Transformers for the improvement of text representations. In particular, for the identification of AL, transfer learning has been applied considering different pre-trained models, such as ELMO, GPT-2 and BERT (Liu et al., 2019; Nikolov and Radivchev, 2019).

Regarding the classification stage, a vast range of approaches and techniques have also been proposed. These approaches could be divided into two main categories; the first category relies on traditional classification algorithms such as Naive Bayes, Support Vector Machines (SVM), Logistic Regression and Random Forest (Burnap and Williams, 2016; Nobata et al., 2016; Davidson et al., 2017; Schmidt and Wiegand, 2017). On the other hand, the second category includes DL approaches, which rely on the use of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), to accomplish the tasks of feature extraction (Badjatiya et al., 2017; Gambäck and Sikdar, 2017) and dependency learning (Badjatiya et al., 2017; Saksesi et al., 2018). In addition to this, the combination of both types of Neural Networks have been used for the development of powerful structures that capture order information between the extracted features (Zhang et al., 2018; Amrutha and Bindu, 2019).

Finally, most recent works in abusive AL identification have considered DL architectures with the addition of an AM. One of the first works introducing attention into the task used the SA

mechanism to detect abuse in portal news and Wikipedia (Pavlopoulos et al., 2017). Subsequently, (Chakrabarty et al., 2019) showed that the use of CA introduced by (Yang et al., 2016) improved the results of SA in this task. Later in (Jarquín-Vásquez et al., 2020) the use of the CA is extended at a word n-grams level, showing the advantages in the usage of word sequences when identifying AL. Regarding other tasks outside the AL identification, some hybrid AMs have been proposed for the combination and representation of different instances and modalities (Khullar and Arora, 2020; Zhang et al., 2020), unlike these hybrid approaches, the proposed SCA mechanism combines the features of the SA and CA mechanisms at an instance level. Motivated by these previous works and with the goal of creating an AM that handles both, the internal and external relationships between words, in this paper we propose the SCA mechanism.

3 Self-contextualized attention

This section is divided into two subsections. First we introduce our proposed SCA mechanism, which is designed to be applied to any sequence of encoding features. Subsequently, we present the DNN architecture used as our classification framework. For more details related to the AMs, we refer the reader to the following work: (Chaudhari et al., 2020).

3.1 Self-contextualized attention mechanism

Given a sequence of encoding features $H = \{h_1, h_2, ..., h_n\}$, where $H \in \mathbb{R}^{k \times n}$, k is the number of the encoding features and h_i refers to the $i\text{-}th$ element of H, the purpose of our proposed SCA mechanism is to generate a global context-aware representation G, that considers both the internal and external relationships between the encoding features of H. Figure 1 shows the general architecture of our proposed SCA mechanism. This architecture is divided into three major stages, each of them is illustrated by the 3 rectangles, corresponding to the SA, CA and SCA stages. Below, we present in detail the aforementioned stages.

SA stage: as in (Pavlopoulos et al., 2017) the main purpose of SA is the building of connections within the elements of the same sequence, but at different positions. The use of SA allows the modeling of both long-range and local dependencies, this is captured by the attention filter $\alpha_s \in \mathbb{R}^{n \times n}$ defined in the Equation 1. This attention filter is

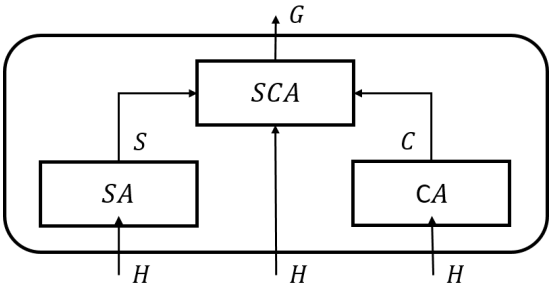

Figure 1: Proposed self-contextualized attention mechanism.

calculated with the dot product similarity between all the pairs of elements of H, later these values are smoothed with the use of a softmax function. Finally, the context-aware representation $S \in \mathbb{R}^{k \times n}$ shown in the Equation 2, is calculated with the matrix multiplication of H and α_s^T, where α_s is used to highlight and filter out the most and less relevant encoding features, respectively.

$$\alpha_s = softmax(H^T \cdot H) \tag{1}$$

$$S = H\alpha_s^T \tag{2}$$

CA stage: unlike the previous stage, the CA mechanism uses a context vector $u_h \in \mathbb{R}^k$, which is randomly initialized and jointly learned during the training process, this vector is used as a query vector in order to obtain the attention values $\alpha_c \in \mathbb{R}^n$ by measuring the similarity between the elements of the sequence H and the application domain represented by u_h. This similarity is calculated in the Equation 3 by calculating the scalar dot product of u_h^T and H; the resulting values are smoothed with the use of a softmax function. Contrasting the CA mechanism proposed by (Yang et al., 2016), instead of using a weighted sum between each attention value and its corresponding encoding features for the final sequence representation, our context-aware representation $C \in \mathbb{R}^{k \times n}$ shown in Equation 4, takes all the information of the attention values, by doing an element-wise multiplication $\odot$, within each scalar of α_c and its corresponding encoding features h_i.

$$\alpha_c = softmax(u_h^T \cdot H) \tag{3}$$

$$C = \alpha_c \odot H \tag{4}$$

SCA stage: since the previous stages generate two different context-aware representations S and

C, respectively. The purpose of this stage is to merge these representations in order to create a global context-aware representation $G \in \mathbb{R}^{k \times n}$ that integrates both, the internal and external relationships. These relationships are captured with the global attention filter $\alpha_g \in \mathbb{R}^{n \times n}$, which is calculated by the smoothed dot product similarity between S and C, as shown in Equation 5. This attention filter can be seen as a high level attention representation, since it is calculated based on the local dependencies and the application domain. Finally, the global context-aware representation G is calculated in Equation 6 with the matrix multiplication of H and α_g^T.

$$\alpha_g = softmax(S^T \cdot C) \tag{5}$$

$$G = H\alpha_g^T \tag{6}$$

The proposed SCA mechanism can be applied to any sequence of encoding features H. For the purposes of this work, each element of the sequence is represented by the word encoding features h_i.

3.2 Classification framework

In order to integrate our proposed SCA mechanism into the AL identification task, we adapt a modular and well-performing DNN architecture, as our classification framework. This architecture was presented in (Yang et al., 2016; Chakrabarty et al., 2019) and its designed to modularly manage different AM. The adapted architecture is shown in Figure 2; it consists of four main stages, which are described below.

The first and second stages correspond to the input and encoding stages, respectively. The *input stage* is integrated by the embedding matrix $X \in \mathbb{R}^{d \times n}$, which is represented by a sequence of n d-dimensional word vectors x_i. Subsequently, the embedding matrix X passes as input to the *encoding stage*, which is conformed by a Bidirectional Gated Recurrent Unit (Bi-GRU) layer. The Bi-GRU layer accomplish the sequence encoding task by summarizing the information of the whole sequence X centered around each word annotation; the producing encoding stage generates a sequence of encoding features $H \in \mathbb{R}^{k \times n}$.

Since not all words contribute equally for the meaning and representation of a sequence, the *third stage* corresponds to the attention stage, including the SCA mechanism and the average pooling layer. Specifically, the sequence encoded features H are passed as input to the SCA mechanism, which generates a global context-aware representation G; since the next stage uses a vector for the classification layers, the matrix G is reduced with the average pooling layer, generating a high level representation vector $g \in \mathbb{R}^k$, which summarizes the most relevant information from G. Finally, the *Fourth stage* uses the representation vector g as input for the classification layers; two layers handle the final classification, a dense layer with a Rectified Linear Unit (ReLU) activation function, and a fully-connected softmax layer to obtain the class probabilities and get the final classification. The implementation details and the hyperparameter settings are presented in Section 4.2.

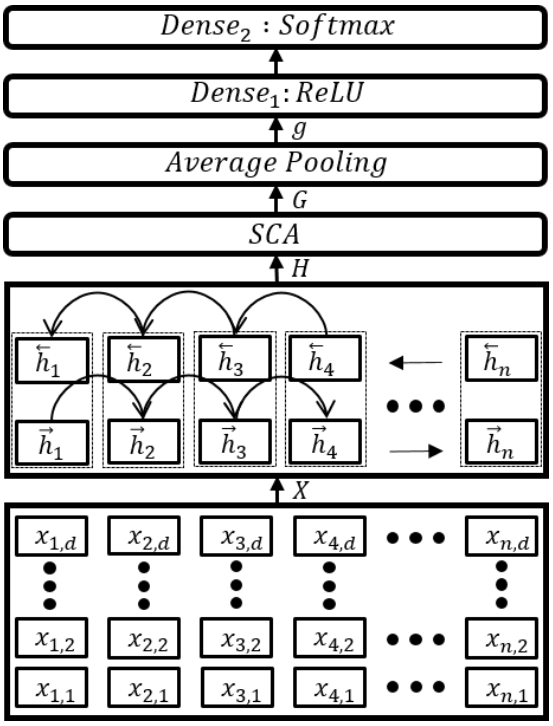

Figure 2: Adapted classification framework, based on a DNN architecture.

4 Experimental settings

This section presents the experimental settings. First, we introduce the four evaluation datasets, which correspond to Twitter collections. Then, with the purpose of facilitating the replicability of our results, we present our method's implementation details, starting from the text preprocessing phase, up to the configuration of the classification framework.

4.1 Datasets for AL identification

AL can be of different types, its main divisions are distinguished by the target and severity of the insults. Accordingly, different collections and evaluation campaigns have considered different kinds of AL for its study. Below we present a brief description of the four English datasets we used in our experiments. From now on we will refer to them as DS1, DS2, DS3, and DS4.

DS1 (Davidson et al., 2017) and DS2 (Zeerak and Dirk, 2016) were some of the first large-scale datasets for abusive tweet detection; DS1 focuses on the identification of racist and sexist tweets, whereas DS2 focuses on identifying tweets with abusive language and hate speech. On the other hand, DS3 (Marcos et al., 2019) and DS4 (Fersini et al., 2018) were used in the *SemEval-2019 Task 6*, and in the *Evalita 2018* Task on Automatic Misogyny Identification (AMI) respectively. DS3 focuses on identifying offensive tweets, whereas DS4 focuses on identifying misogyny in tweets. Both shared tasks provide a fine-grained evaluation through different sub-tasks; in this work, we focus on the sub-task A (binary classification of offenses and misogyny, respectively).

Figure 3 resumes the information about the classes distribution of the four collections.

4.2 Implementation details

Different text preprocessing operations were applied: user mentions and links were replaced by the default tokens <*user*> and <*url*>; in order to enrich the vocabulary, all hashtags were segmented by words (*e.g.* #BuildTheWall - build the wall) with the use of the ekphrasis library, proposed in (Baziotis et al., 2017); in addition to this, all emojis were converted into words (*e.g.* ☺ - smiley face) using the demoji[2] library; stop words were removed, with the exception of personal pronouns; all text was lowercased and non-alphabetical characters as well as consecutive repeated words were removed. For word representation we used pre-trained fastText embeddings (Mikolov et al., 2018), trained with subword information on Common Crawl, which have been recognized as useful for this task according to the study presented in (Corazza et al., 2020).

Table 1 presents the hyperparameter settings of the adapted DNN. The network was trained for a total of 15 epochs, with a learning rate of 1e–4, using the Adam optimizer (Kingma and Ba, 2015) and a Dropout rate of 15%. In order to compare the robustness of our proposal, we consider four baseline architectures: the first architecture is based on a simple Bi-GRU network, which receives words as input but does not use any attention layers; the second and third architectures employ the same Bi-GRU network with the addition of a SA and CA layer, respectively; finally, in order to compare the performance of our proposed SCA mechanism against a novel AL identification approach, the fourth baseline is based on a fine-tuned BERT[3] base model (12 layers, 768 hidden size, 12 attention heads per layer), built with the addition of the task-specific inputs and the end-to-end fine-tuning of all parameters. As described in (Devlin et al., 2019), we take the last layer encoding of the classification token <CLS> and use it as input for the softmax classification layer. These four baselines architectures and our classification framework are referred in the experiments as: $Bi-GRU$, $Bi-GRU_{SA}$, $Bi-GRU_{CA}$, $BERT_{BASE}$, and $Bi-GRU_{SCA}$, respectively. It is important to mention that the first three baseline architectures used the same hyperparameter settings.

Vectors and Variables		**Size**
n		50
d		300
k, u_h		128
Layer	**Input size**	**Output size**
Embedding	50	50x300
Bi-GRU	50x300	50x128
SCA	50x128	50x128
Avg Pooling	50x128	128
Dense_1	128	64
Dense_2	64	#Classes

Table 1: DNN architecture hyperparameters.

5 Experimental results

This section is organized in three subsections. Sections 5.1 and 5.2 present the quantitative results of the experimentation, corresponding to the comparison of our proposed SCA mechanism against the baselines and state-of-the-art results. Finally, Section 5.3 presents some qualitative results of the SCA mechanism, through the analysis and visualization of the attention values.

[2] https://pypi.org/project/demoji/

[3] https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-12_H-768_A-12/1

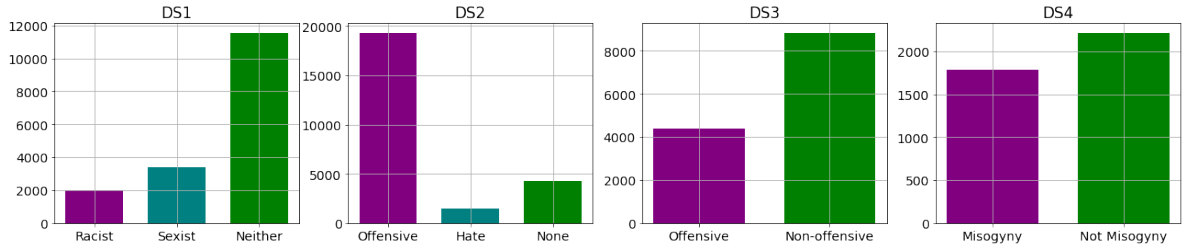

Figure 3: The classes distribution of the four used datasets.

5.1 Quantitative effectiveness of the SCA mechanism

Table 2 shows the results of the mean and standard deviation corresponding to the 10-fold cross validation evaluation applied to our classification framework DNN architecture ($Bi-GRU_{SCA}$), as well as the four baselines simplified architectures $Bi-GRU$, $Bi-GRU_{SA}$, $Bi-GRU_{CA}$ and $BERT_{BASE}$. For sake of comparison, we evaluate all the collections using the macro-average F_1 score, which is commonly used in the AL identification task.

Centering the analysis of results on the first three baselines and on our classification framework (columns 2 - 5), the results indicate that the use of AM outperformed the base Bi-GRU network (column 2 vs columns 3 - 5) by at least a margin of 1.1%. In addition, the use of the CA outperformed the use of SA (column 4 vs column 3) by at least a margin of 1.2%, which is consistent according to the results obtained in (Chakrabarty et al., 2019). Finally, comparing the use of our proposed SCA mechanism against the use of SA and CA (column 5 vs columns 3 and 4), better results are obtained in the four evaluation datasets, improving the results by at least a margin of 1.1%. Since the use CA baseline outperforms the SA based one, we compared $Bi-GRU_{SCA}$ vs $Bi-GRU_{CA}$ with the Chi Squared Test, obtaining statistically significant values with $p \leq 0.001$.

Table 2 also compares the results from our proposed SCA mechanism with respect to the $BERT_{BASE}$ baseline (column 5 vs column 6). It is shown that the $Bi-GRU_{SCA}$ DNN obtained better results in 3 out of 4 datasets. In addition to the outstanding results, the use of our $Bi-GRU_{SCA}$ DNN has a considerably lower number of parameters compared to the $BERT_{BASE}$ model (110M vs 7M), which greatly reduces the computing power necessary to run our DNN. Finally, compared to some novel approaches for the AL identification task (Alshaalan and Al-Khalifa, 2020), our DNN improves the model interpretability, through the SCA mechanism.

5.2 Comparison with the state-of-the-art

In this subsection we compare our proposed DNN architecture ($Bi-GRU_{SCA}$) with state-of-the-art approaches. Since the datasets DS1 and DS2 are presented as a single dataset, in order to have a fair comparison with other works, these were partitioned into 80% for training, 10% for validation and 10% for testing, in addition, the weighted-average F_1 score was used as an evaluation measure for these datasets. In the case of DS3 and DS4 datasets, the partitions corresponding to the training and testing were considered for the evaluation; since these datasets come from shared tasks, the evaluation measures were adjusted to each of them, specifically, DS3 and DS4 were evaluated using the macro-average F_1 score and the accuracy, respectively.

Table 3 presents the results of our proposed $Bi-GRU_{SCA}$ DNN architecture in comparison with state-of-the-art results. It shows that the $Bi-GRU_{SCA}$ DNN obtained better results in 2 out of 4 datasets. It is important to note that the state-of-the-art results from the DS2 and DS3 datasets only improved our results by margin of 1% and 0.03%, respectively. Specifically, in (Mozafari et al., 2019), which corresponds to the DS1 and DS2 state-of-the-art results, the use of a BERT-based CNN is implemented for the feature extraction of the transformer encoders, generating a hierarchical encoded vector, used for the AL classification.

Regarding the state-of-the-art results from the DS3 and DS4 datasets, the best performance teams corresponding to each shared task were considered, on the one hand, *NULI* the best performance team in the DS3 shared task (Liu et al., 2019), used a BERT-base-uncased model with default-parameters, using a max sentence length of 64 and a variety of text pre-processing techniques, on the

Dataset	$Bi-GRU$	$Bi-GRU_{SA}$	$Bi-GRU_{CA}$	$Bi-GRU_{SCA}$	$BERT_{BASE}$
DS1	0.7614 ±0.0083	0.8162 ±0.0079	0.8271 ±0.0069	**0.8378 ±0.0082**	0.8291 ±0.0076
DS2	0.7438 ±0.0072	0.7721 ±0.0081	0.7874 ±0.0074	0.7984 ±0.0078	**0.8052 ±0.0083**
DS3	0.7698 ±0.0081	0.8052 ±0.0078	0.8247 ±0.0085	**0.8423 ±0.0064**	0.8398 ±0.0081
DS4	0.6541 ±0.0096	0.6654 ±0.0073	0.6782 ±0.0067	**0.6937 ±0.0086**	0.6906 ±0.0076

Table 2: Comparison results from the four baselines architectures and our classification framework in four datasets for AL identification (*all the collections were evaluated with the macro-average F_1).*

Dataset	$Bi-GRU_{SCA}$	*state-of-the-art*	*Reference*
DS1	**0.89**	0.88	(Mozafari et al., 2019)
DS2	0.91	**0.92**	(Mozafari et al., 2019)
DS3	0.826	**0.829**	(Liu et al., 2019)
DS4	**0.738**	0.704	(Saha et al., 2018)

Table 3: Comparison results from our classification framework and state-of-the-art approaches in four datasets for AL identification (*DS1 and DS2 were evaluated with the weighted-average F_1, DS3 and DS4 were evaluated using the macro-average F_1 and the accuracy, respectively).*

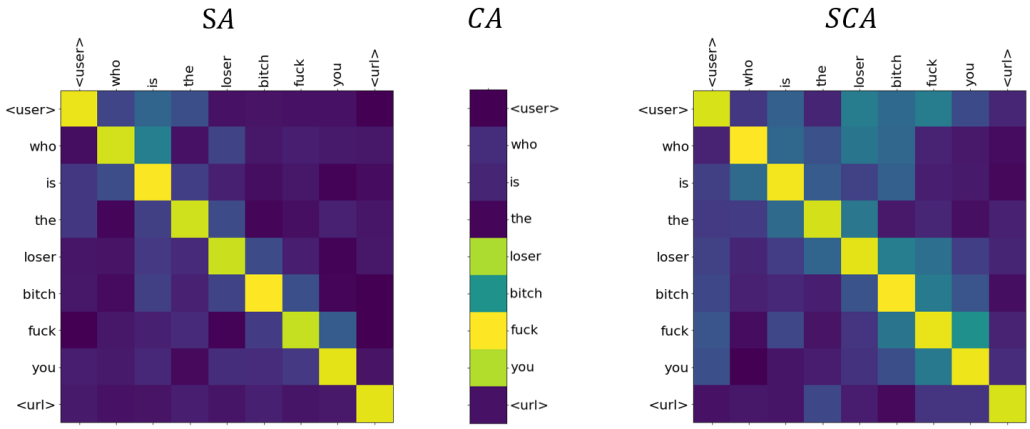

Figure 4: Attention heatmaps visualization, corresponding to the α_s, α_c, and α_g attention filter values. The Example shown in the attention heatmaps was taken from the DS3 dataset.

other hand, *hateminers* achieved the highest performance on the DS4 shared task (Saha et al., 2018), with a run based on a vector representation that concatenates sentence embedding, TF-IDF and average word embeddings coupled with a Logistic Regression model. Unlike the reported state-of-the-art approaches, the use of our SCA mechanism on a simple and well-performed DNN, obtains competitive results, without the use of complex DNN (Mozafari et al., 2019), or large amounts of resources and features (Saha et al., 2018).

The boxplot graphs shown in Figure 5, compares our $Bi-GRU_{SCA}$ performance results (red rhombus) against the top-10 results corresponding to the shared tasks *SemEval 2019 Task 6* and *AMI Evalita 2018*, respectively. As shown in the graphs, our results are competitive with respect to the top-10 results obtained by the best participating teams in each sub-task A. In both boxplot graphs our results remain above the third quartile, specifically, in the *AMI Evalita 2018* shared task an outstanding performance is obtained with the use of our proposed SCA mechanism in the classification framework.

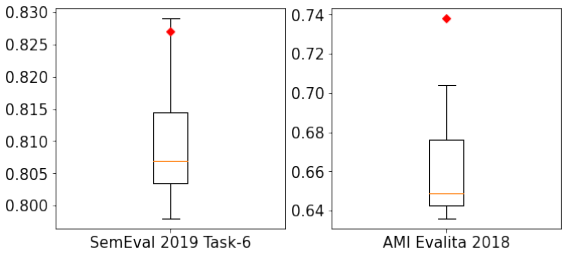

Figure 5: Comparative Boxplot graphs from our results (red rhombus) vs. the top-10 results of the shared tasks.

5.3 Qualitative effectiveness of the SCA mechanism

NOTE: This subsection contains examples of language that may be offensive to some readers, these do not represent the perspectives of the authors.

In order to understand the effectiveness of our proposed SCA mechanism in the improvement of the sequences representation, this subsection presents the qualitative results of the analysis and visualization of the attention values. Since the SCA mechanism integrates both, the SA and CA mechanisms, the attention values were considered at these three different levels, with the analysis of the α_s, α_c and α_g attention filters, which correspond to the SA, CA and SCA mechanisms.

Figure 4 shows the visualization of the attention heatmaps corresponding to the three attention filters values integrated by the SCA mechanism. The example shown in the figure *"<user> who is the loser bitch fuck you <url>"* corresponds to an offensive instance taken from the DS3 dataset. As shown in the figure, the values of the attention filter α_s, corresponding to the SA, tend to be more relevant with respect to their own elements and their closest neighbors, for example, in the case of the most relevant words to *"who"*, the same word *"who"* is found, followed by the word *"is"*, likewise, in the case of the most relevant words to *"fuck"*, the words *"fuck"*, *"you"* and *"bitch"* are found. On the other hand, the values of the attention filter α_c, corresponding to the CA, indicate the most relevant words for the AL identification; as can be seen in the central heatmap from the Figure 4, the most relevant words are: *"loser"*, *"bitch"* and *"fuck"*, which indeed correspond to words potentially used in offensive contexts.

Finally, the values of the attention filter α_g, corresponding to the SCA, are shown in the right heatmap from Figure 4. The attention filter α_g shows the combination of both AM, which improves the representation of an instance. For example, in the produced visualization from the most relevant words to *"<user>"*, a closer relationship to offensive words is now presented, highlighting the words: *"loser"*, *"bitch"* and *"fuck"*, which are often used to offend, something similar is presented with the words *"who"* and *"is"*. On the other hand, the words *"fuck"*, *"you"* and *"bitch"*, in addition to having a better relationship with other offensive words as *"loser"*, are also related to the target of the offense: *"<user>"*.

6 Conclusions and future work

One of the main problems in the use of current AMs is the loss of contextual or internal information between the elements of a sequence. To tackle this issue we proposed the SCA mechanism, which integrates the SA and CA mechanisms for the construction of a representation that considers both, the internal and contextual relationships between the elements of a sequence. Due to the highly context-dependent interpretation of words in the AL identification, in this work we explore the use of the proposed SCA mechanism in the AL identification. The results obtained in four collections, considering different kinds of AL, were encouraging; they improved state-of-the-art approaches in 2 out of 4 datasets. In addition to this, the SA and CA mechanisms were evaluated against the SCA mechanism, the results show a quantitative and qualitative improvement in the use of the SCA mechanism, which allowed concluding that the use of the SCA mechanism is useful for discriminating between offensive and non-offensive contexts.

Since the most recent approaches are based on Transformers, as future work we plan to explore the use of our proposed SCA mechanism in the design of a multi-head SCA architecture. Additionally, we consider exploring new ways of combining the SA and CA mechanisms, as well as some novel approaches in the building of the SCA mechanism without the need of computing the SA and CA mechanisms individually. Finally, we consider the application of the proposed SCA mechanism in other related tasks where the interpretation of words is highly context dependent such as the detection of deception or the detection of depressed social media users.

Acknowledgements

We thank CONACyT-Mexico for partially supporting this work under project grant CB-2015-01-257383 and scholarship 925996.

References

Raghad Alshaalan and Hend Al-Khalifa. 2020. Hate speech detection in saudi twittersphere: A deep learning approach. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 12–23, Barcelona, Spain (Online). Association for Computational Linguistics.

B. R. Amrutha and K. R. Bindu. 2019. Detecting hate speech in tweets using different deep neural network

architectures. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 923–926.

Mario Ezra Aragón, Horacio Jesús Jarquín-Vásquez, Manuel Montes-y-Gómez, Hugo Jair Escalante, Luis Villaseñor Pineda, Helena Gómez-Adorno, Juan Pablo Posadas-Durán, and Gemma Bel-Enguix. 2020. Overview of MEX-A3T at iberlef 2020: Fake news and aggressiveness analysis in mexican spanish. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020) co-located with 36th Conference of the Spanish Society for Natural Language Processing (SEPLN 2020), Málaga, Spain, September 23th, 2020*, volume 2664 of *CEUR Workshop Proceedings*, pages 222–235. CEUR-WS.org.

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*, Republic and Canton of Geneva, CHE. 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, Vancouver, Canada. Association for Computational Linguistics.

Pete Burnap and Matthew L. Williams. 2016. Us and them: identifying cyber hate on twitter across multiple protected characteristics. *EPJ Data Sci.*, 5:11.

Noé Cecillon, Vincent Labatut, Richard Dufour, and G. Linarès. 2019. Abusive language detection in online conversations by combining content- and graph-based features. *Frontiers in Big Data*, 2.

Tuhin Chakrabarty, Kilol Gupta, and Smaranda Muresan. 2019. Pay "attention" to your context when classifying abusive language. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 70–79. Association for Computational Linguistics.

Sneha Chaudhari, Gungor Polatkan, R. Ramanath, and Varun Mithal. 2020. An attentive survey of attention models. *Association for Computing Machinery*, 37.

Michele Corazza, Stefano Menini, Elena Cabrio, Sara Tonelli, and Serena Villata. 2020. A multilingual evaluation for online hate speech detection. *ACM Transactions on Internet Technology*, 20(2).

Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, Montréal, Québec, Canada, May 15-18, 2017*, pages 512–515. AAAI Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Elisabetta Fersini, Debora Nozza, and Paolo Rosso. 2018. Overview of the evalita 2018 task on automatic misogyny identification (AMI). In *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it)*, volume 2263 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Björn Gambäck 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. Association for Computational Linguistics.

Sameer Hinduja and Justin W. Patchin. 2010. Bullying, cyberbullying, and suicide. *Archives of Suicide Research*, 14(3):206–221.

Horacio Jesús Jarquín-Vásquez, Manuel Montes-y Gómez, and Luis Villaseñor-Pineda. 2020. Not all swear words are used equal: Attention over word n-grams for abusive language identification. In *Pattern Recognition*, pages 282–292, Cham. Springer International Publishing.

Aman Khullar and Udit Arora. 2020. MAST: Multimodal abstractive summarization with trimodal hierarchical attention. In *Proceedings of the First International Workshop on Natural Language Processing Beyond Text*, pages 60–69. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Ritesh Kumar, Aishwarya N. Reganti, Akshit Bhatia, and Tushar Maheshwari. 2018. Aggression-annotated corpus of Hindi-English code-mixed data. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ping Liu, Wen Li, and Liang Zou. 2019. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91. Association for Computational Linguistics.

Zampieri Marcos, Malmasi Shervin, Nakov Preslav, Rosenthal Sara, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86. Association for Computational Linguistics.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA).

Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. 2019. A bert-based transfer learning approach for hate speech detection in online social media. In *Complex Networks and Their Applications VIII - Volume 1 Proceedings of the Eighth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2019, Lisbon, Portugal, December 10-12, 2019*, volume 881 of *Studies in Computational Intelligence*, pages 928–940. Springer.

Alex Nikolov and Victor Radivchev. 2019. Nikolov-radivchev at SemEval-2019 task 6: Offensive tweet classification with BERT and ensembles. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 691–695, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Chikashi Nobata, 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*, page 145–153. International World Wide Web Conferences Steering Committee.

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*, pages 1125–1135, Copenhagen, Denmark. Association for Computational Linguistics.

Punyajoy Saha, Binny Mathew, Pawan Goyal, and Animesh Mukherjee. 2018. Hateminers : Detecting hate speech against women. *CoRR*, abs/1812.06700.

Arum Sucia Saksesi, M. Nasrun, and C. Setianingsih. 2018. Analysis text of hate speech detection using recurrent neural network. In *2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, pages 242–248.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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 and Hovy Dirk. 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. Association for Computational Linguistics.

Dongxiang Zhang, Yuyang Nie, Sai Wu, Yanyan Shen, and Kian-Lee Tan. 2020. Multi-context attention for entity matching. In *Proceedings of The Web Conference 2020*, WWW '20, page 2634–2640. Association for Computing Machinery.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *The Semantic Web*, pages 745–760, Cham. Springer International Publishing.