Detect All Abuse! Toward Universal Abusive Language Detection Models

Kunze Wang[1*], Dong Lu[1*], Soyeon Caren Han[2], Siqu Long[1], Josiah Poon[2]**
The University of Sydney, Sydney, Australia
[1]{kwan4418,dolu4031,slon6753}@uni.sydney.edu.au
[2]{caren.han,josiah.poon}@sydney.edu.au

Abstract

Online abusive language detection (ALD) has become a societal issue of increasing importance in recent years. Several previous works in online ALD focused on solving a single abusive language problem in a single domain, like Twitter, and have not been successfully transferable to the general ALD task or domain. In this paper, we introduce a new generic ALD framework, MACAS, which is capable of addressing several types of ALD tasks across different domains. Our generic framework covers multi-aspect abusive language embeddings that represent the target and content aspects of abusive language and applies a textual graph embedding that analyses the user's linguistic behaviour. Then, we propose and use the cross-attention gate flow mechanism to embrace multiple aspects of abusive language. Quantitative and qualitative evaluation results show that our ALD algorithm rivals or exceeds the six state-of-the-art ALD algorithms across seven ALD datasets covering multiple aspects of abusive language and different online community domains. The code can be downloaded from https://github.com/usydnlp/MACAS.

1 Introduction

Abusive language in online communities has become a significant societal problem (Nobata et al., 2016) and online abusive language detection (ALD) aims to identify any type of insult, vulgarity, or profanity that debases a target or group online. It is not only limited to detecting offensive language (Razavi et al., 2010), cyberbullying (Xu et al., 2012), and hate speech (Djuric et al., 2015), but also to more nebulous or implicit forms of abuse. Many social media companies and researchers have utilised multiple resources, including machine learning, human reviewers and lexicon-based text analytics to detect abusive language (Waseem, 2016; Qian et al., 2018). However, none of them can perfectly resolve the ALD task because of the difficulties of moderating user content and in classifying ambiguous posts (Metz and Issac, 2019). On the technical side, previous ALD models were developed on only a few subtasks (e.g. hate speech, racism, sexism) in a single domain (like Twitter), and each specialised model is not successfully transferable to general ALD in different online communities.

Our research question is, "What would be the best generic ALD model that can be used for different types of abusive language detection sub-tasks and in different online communities?" To solve this, we found that Waseem et al. (2017) reviewed the existing online abusive language detection literature, and defined a generic abusive language typology that can encompass the targets of a wide range of abusive language subtasks in different types of domain. The typology is categorised in the following two aspects: **1) Target aspect**: The abuse can be directed towards either a) a specific individual/entity or b) a generalised group. This is an essential sociological distinction as the latter refers to a whole category of people, like a race or gender, rather than a specific individual or organisation; **2) Content aspect**: The abusive content can be explicit or implicit. Whether directed or generalised, explicit abuse is unambiguous in its potential to be damaging, while implicit abusive language does not immediately

---

Dataset	Source	Size	Composition
Waseem(Waseem and Hovy, 2016)	Twitter	16.2k	*Racism(11.97%), Sexism(19.43%), None(68.60%)*
HatEval(Basile et al., 2019)	Twitter	13k	*Hateful(42.08%), Non-hateful(57.92%)*
OffEval(Zampieri et al., 2019)	Twitter	13.2k	*Offensive(33.23%), Not-offensive(66.77%)*
Davids(Davidson et al., 2017)	Twitter	24.8k	*Hate(5.77%), Offensive(77.43%), Neither(16.80%)*
Founta(Founta et al., 2018)	Twitter	99k	*Abusive(27.15%), Hateful(4.97%), Normal(53.85%), Spam(4.97%)*
FNUC(Gao and Huang, 2017)	Fox News Discussion Threads	1.5k	*Hateful(28.50%), Non-hateful(71.50%)*
StormW(de Gibert et al., 2018)	Stormfront(forum)	10.7k	*Hate(10.93%), NoHate(89.07%)*

Table 1: Comparison and Statistical analysis of seven benchmark datasets evaluated in this paper. The composition column represents different class aspects, and the class distribution in each dataset.

imply abuse (through the use of sarcasm, for example). For example, assume that we have a tweet "F***". "You are sooo sweet like other girls". It includes all those aspects; the directed target ("yourself"), the generalised target ("girls"), the explicit content ("F***"), and the implicit content ("You are sooo sweet").

Inspired by this abusive language typology, we propose a new generic ALD framework, MACAS (**M**ulti-**A**spect **C**ross **A**ttention **S**uper Joint for ALD), using aspect models and a cross-attention aspect gate flow. First, we build four different types of abusive language aspect embeddings, including directed target, generalised target, explicit content, and implicit content. We also propose to use a heterogeneous graph to analyse the linguistic behaviour of each author and learn word and document embeddings with graph convolutional networks (GCNs). Not every online community (e.g. news forums) allows user-to-user relationship (e.g. follower-following), so we avoid using user-community relationship information. Then, we propose a cross-attention aspect gate flow to obtain the mutual enhancement between the two aspects. The gate flow contains two gates, target gate and content gate, then fuses the outputs of those gates. The target gate draws on the content probability distribution, utilising the semantic information of the whole input sequence along with the target source, while the content gate takes in the target aspect probability distribution as supplementary information for content-based prediction. For evaluation, we test six state-of-the-art ALD models across seven datasets focused on different aspects and collected from different domains. Our proposed model rivals or exceeds those ALD methods on all of the evaluated datasets. The contributions of the paper can be summarised as follows: 1) We perform a rigorous comparison of six state-of-the-art ALD models across seven ALD benchmark datasets, and find those models do not embrace different types of abusive language aspects in different online communities. 2) We propose a generic new ALD algorithm that enables explicit integration of multiple aspects of abusive language, and detection of generic abusive language behaviour in different domains. The proposed model rivals state-of-the-art algorithms on ALD benchmark datasets and performs best overall.

2 Related Work

2.1 ALD Datasets

We briefly review the seven ALD benchmark datasets (Table 1), which were collected from different online community sources and focused on multiple compositions. **Waseem** (Waseem and Hovy, 2016) is a Twitter ALD dataset regarding the specific aspects of racist and sexist. The collected tweets were labeled into *Racism*, *Sexism* or *None*. **HatEval** (Basile et al., 2019) is a Twitter-based hate speech detection dataset released in SemEval-2019. It provides a general-level hate speech annotation, *Hateful* or *Non-hateful*, especially against immigrants and women. **OffEval** (Zampieri et al., 2019) covers the Twitter-based offensive language detection task in SemEval-2019. It annotates as *Offensive* or *Not-offensive*, and includes insults, threats, and any form of untargeted profanity. **Davids** (Davidson et al., 2017) is a Twitter-based ALD dataset, which includes three classes, *Hate*, *Offensive* or *Neither* based on the hate speech lexicon from *Hatebase.org*. **Founta** (Founta et al., 2018) is a large Twitter-based ALD dataset claimed to be annotated with high accuracy based on their proposed incremental and iterative annotation method. It is annotated with four classes, *Hateful*, *Abusive*, *Normal* or *Spam*. **FNUC** (Gao and Huang, 2017) is a hate speech detection dataset, which was collected from complete Fox News discussion threads, and annotated with the general level categories *Hateful* or *Non-hateful*. **StormW**(de Gibert et al., 2018) is a Stormfront-based hate speech detection dataset with general-level labels *Hate* and *NoHate*. Stormfront is a supremacist forum where people promote white nationalism and antisemitism.
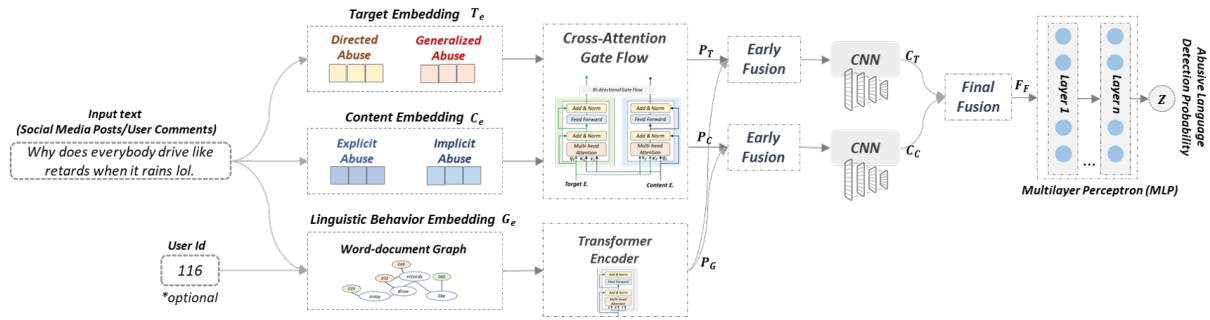
Figure 1: The conceptual architecture of our model *MACAS*

2.2 ALD Approaches

In the early stages, ALD was commonly addressed via hand-crafted rules and manual feature engineering. The first reported ALD work(Spertus, 1997) utilised a decision tree to detect hostile messages based on heuristic rules. Yin et al. (2009) and Razavi et al. (2010) added lexicon-based features together with semantic rules and designed a linear SVM and Naïve Bayes classifier for detecting hostile language. Djuric et al. (2015) first applied in ALD neural networks with the paragraph2vec (Le and Mikolov, 2014) representation. Nobata et al. (2016) introduced a Yahoo! dataset and tested it with neural networks by applying a combination of word, character-based and syntactic features. Recently, deep learning techniques have become popular in ALD. Badjatiya et al. (2017) tested FaxtText/Glove, Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTMs) in detecting hate speech. Park and Fung (2017) designed a HybridCNN (word-level and character-level) model on abusive tweet detection in both one-step and two-step style. Several works have applied bidirectional Gated Recurrent Unit (Bi-GRU) networks with Latent Topic Clustering (LTC) Lee et al. (2018) and a a transformer-based framework Bugueño and Mendoza (2019). Some works integrated user profiling into their ALD models. Qian et al. (2018) utilised the bi-LSTM to model the historical behaviour of users to generate inter-user and intra-user representation. Mishra et al. (2018) applied node2vec (Grover and Leskovec, 2016) to the constructed community graph of users to derive the user embedding. However, a user profiling-based approach is only possible when the user profiles are public and when the domain provides the user-community relation information.

3 The MACAS ALD Model

We propose the **M**ulti-**A**spect **C**ross **A**ttention **S**uper Joint model for ALD. It is designed as an generic ALD that can embrace different types of abusive language aspects in different online communities. As shown in Figure 1, MACAS can be divided into three main phases: 1) **Multi-Aspect features embedding**[Sec.3.1]. The Multi-Aspect Embedding Layer represents understanding of multi-aspects of abusive language for detecting generic abusive language behaviours. We focus on two main aspects, target and content, and each aspect has two sub-aspects. **1) Target aspect** represents abuse directed towards either a) a specific individual/entity or b) a generalised group (e.g. gender or race). **2) Content aspect** covers a) explicit or b)implicit. Explicit abuse is unambiguous in its potential to be damaging, while implicit abusive language does not immediately impact (e.g. sarcasm). In addition to this, if the platform provides users' historical posts, we apply Graph Convolutional Network(GCN)s to build a word-document graph embedding that represents linguistic behaviours of users. Not every online community (e.g. news forums) has user-to-user relationships (e.g. follower-following), so we avoid using user-community relationship and community network information. 2) **Cross-Attention Gate Flow for integrating multi-aspects** [Sec.3.2] The Cross-Attention gate produces the joint integration of the target aspect and content aspect model and obtains the mutual enhancement between the two aspects. This is for producing well-integrated multi-aspects and improving the performance of generic ALD. 3) **Final Aggregation of learned ALD embeddings** [Sec.3.3] We aggregate multi-aspect embeddings and the user's linguistic behaviour embedding across the online post using convolutional neural networks, and produce the ALD using multi-layer-perceptron.

3.1 Multi-Aspect Embedding Layer [1]

3.1.1 Target: Directed Abuse Embedding

Directed abuse is abuse towards a specific individual or entity (Waseem et al., 2017). To model this aspect, a named entity recognition (NER) approach is used. To train the NER model, we apply stacked bi-directional LSTMs, which are one of the state-of-the-art models (Chiu and Nichols, 2016). We extract the vector before the final $Softmax$ layer of the NER model and use it as the Directed Abuse Embedding.

3.1.2 Target: Generalised Abuse Embedding

Generalised abuse tends to target people belonging to a small set of categories, primarily gender. The gender debiasing embedding (Kaneko and Bollegala, 2019) is applied. The vocabulary set (V) is split into 4 mutually exclusive sets of words, namely, masculine (V_m), feminine (V_f), neutral (V_n) and stereotypical (V_s). Each word is represented by a vector which is calculated by minimising a loss function to satisfy the criteria: 1) protect the feminine information for words in V_f; 2) protect the masculine information for words in V_m; 3) protect the neutrality for words in V_n (iv) remove gender biases for words in V_s.

3.1.3 Content: Explicit Abuse Embedding

For the explicit abuse, whether the target is directed or generalised, explicit abuse is usually indicated by specific keywords from the homophobic slurs lexicon. We used dict2vec (Tissier et al., 2017), which aims to learn word embeddings based on natural language dictionaries. In this paper, the model is trained by Cambridge, Collins, Oxford, dictionary.com, and we add an abusive language lexicon[2]. This approach first defines strong pairs and weak pairs of words. If both words appear in each other's definition, the word pair is defined as a strong pair. If only one word appears in the other's definition, the word pair is defined as a weak pair. If the words do not appear in each other's definition they are not related. Each word is represented by a vector. Strongly paired words have more similar vectors then weakly paired words which in turn have more similar vectors than unrelated words.

3.1.4 Content: Implicit Abuse Embedding

Implicit abusive language does not immediately imply or denote abuse, similar to sarcasm. Here we use a hybrid of CNN and LSTM-based sarcasm detection models (Ghosh and Veale, 2016). The vector before the final $Softmax$ layer of the sarcasm detection model is the Implicit Abuse Embedding.

#### 3.1.5 Additional: User Linguistic Behaviour Embedding

We model the graph by setting each comment in the training set as a document. The vocabulary is the set of all words in the documents. The corpus is the collection of all documents. The nodes of our graph are the union of the documents and the vocabulary. An edge weighted 1 exists between each node and itself. An edge exists between a document and a word if the word is in that document. The edge is weighted with the TF-IDF for the (document, word) pair, within the corpus. An edge exists between two words if they have a non-negative point-wise mutual information (PMI) with a sliding window size of 20, within the corpus. The weight for the edge is the PMI for the word pair. The edge weightings are compiled into an adjacency matrix combined with the graph's degree matrix and passed into a 2 layer GCN trained to map each document to each user as a label. For datasets without user id provided, we use the actual classification target as the document node label. From this network, we obtain embeddings for each node, that is an embedding of each document or each word. The trained word embeddings G_e are fed into transformer encoders to get linguistic behaviour outputs.

3.2 Cross-Attention Gate Flow

In the Cross-Attention Gate Flow, first, we use a cross transformer encoder for refining our four types of embedding: Directed abuse embedding D, Generalized abuse embedding G, Explicit abuse embedding E and Implicit abuse embedding I. Before putting them into the cross transformer encoders, we combine D

---

[1]In this paper, we use only four state-of-the-art natural language processing techniques that represent each abusive language aspect well. However, we expect that more techniques for each aspect embedding would produce better performance.

[2]http://www.rsdb.org
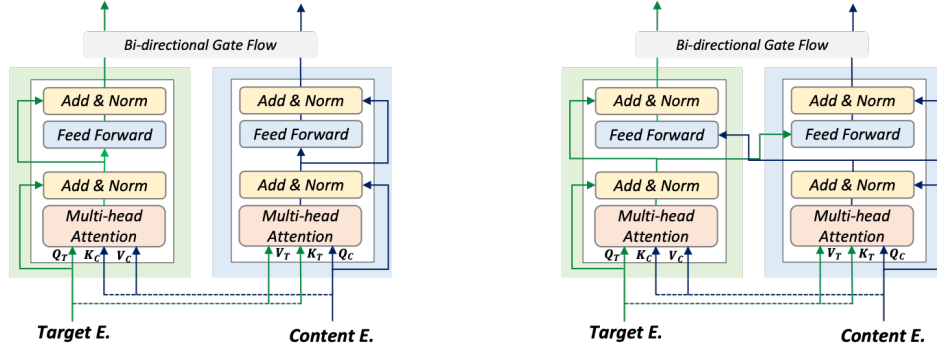
(a) **C**AGF at the **B**eginning   (b) **C**AGF at the **B**eginning and the **M**iddle

Figure 2: Variances of Cross-Attention Gate Flow

with G as Target embedding T_e and broadcast I to sequence length N, them combine it with E as Content embedding C_e. Normally, for the transformer encoder (Vaswani et al., 2017), the attention is calculated using key (K of dimension d_k), query (Q), value (V):

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V. \tag{1}$$

However, to produce the joint integration of target aspect model and content aspect model, we apply the cross-transformer to T_e and C_e. As shown in the Figure 2 for each transformer encoder, we have K,Q,V for T_e and C_e. The K,V of T_e and C_e are switched, which means K,V of T_e goes to the transformer encoder of C_e and K,V of C_e goes to T_e's encoder. Then attention is calculated by

$$Attention_{content} = softmax(\frac{Q_cK_t^T}{\sqrt{d_k}})V_t, \quad Attention_{target} = softmax(\frac{Q_tK_c^T}{\sqrt{d_k}})V_c \tag{2}$$

We call the cross transformer here *Cross at Beginning(CB)*. Similar to the original transformer encoder, each encoder contains one or more encoder stack(s), which mainly consists of two sub-layers: a multi-head attention layer and a fully connected feed-forward neural network (FNN). A residual connection followed by layer normalization is employed around each of the two sub-layers before feeding to the next sub-layer. Another way to produce the joint integration occurs before the FNN layer. The output of Multi-Head Attention will be the input for the FNN layer, and then an Add & Norm layer is applied. Normally, the output of transformer encoder is calculated by

$$Output = norm(FNN(O_{MHA}) + O_{MHA}) \tag{3}$$

The input for FNN can also be switched for Content and Target, which is called *Cross in the Middle (CM)*, the output of transformer encoder will be calculated by

$$T_h = norm(FNN(C_{MHA}) + T_{MHA}), \quad C_h = norm(FNN(T_{MHA}) + C_{MHA}) \tag{4}$$

If the cross happens both at the beginning and in the middle, the structure will be called *Cross at the Beginning and in the Middle (CBM)*. The comparison of different cross transformer structures will be discussed in 5.2. Both of the input embeddings T_e and C_e are of shape $[N, D_e]$, where D_e is the sum of the dimension of the concatenated embedding. The transformer encoder will output T_h and C_h in the same shape $[N, D_e]$. The hidden state of encoders T_h from T_e and C_h from C_e will be used to compute the initial abusive language probability, which is the major input of our bi-directional aspect gate flow.

On top of the Cross-Attention, we introduce the Bi-directional Aspect Gate Flow that contains two gates: content gate and target gate. Denote the input sequences to our gates from the previous layer encoder as $T_h \in \mathbb{R}^{N \times D_T}$ and $C_h \in \mathbb{R}^{N \times D_C}$ where N is the sequence length while D_T and D_C equal to dimension of target embedding and content embedding respectively. In the content gate, we first flatten T_h to be $T_{hf} \in \mathbb{R}^{1 \times (N*D_T)}$. We then pass T_{hf} through a dense layer and apply the $Softmax$ function. The resultant P_{Th} is a D-dimensional probability vector, where $D = N_{cls}$ is the number of distinct labels to classify, $W_C \in \mathbb{R}^{D \times D_C}$ is the weight matrix and $b_C \in \mathbb{R}^{1 \times D}$ is the bias vector. Then we broadcast P_{Th} over N tokens. This yields $\hat{P_{Th}} \in \mathbb{R}^{N \times D}$. Then we concatenate $\hat{P_{Th}}$ with transformer encoder output state C_h from content source, generating the augmented content state $O_C \in \mathbb{R}^{N \times (D+D_C)}$. We then again flatten O_C and pass the output to the dense layer, producing an output matrix $P_C \in \mathbb{R}^{1 \times D}$.

The procedure in the target gate is almost the same as the content gate. Here we flattened the input

sequence C_h, generating the flattened output $C_{hf} \in \mathbb{R}^{1 \times (N*D_C)}$. We then pass the result through a dense layer and apply the $Softmax$ function. The resultant P_{Ch} is also broadcast to be $\hat{P_{Ch}}$ and then concatenated with the target encoder output state T_h, where $O_T \in \mathbb{R}^{N \times (D+D_T)}$ is the augmented target state as output matrix. Finally, O_T is also flattened and then passed to the dense layer, which produces the output matrix $P_T \in \mathbb{R}^{1 \times D}$.

3.3 Final Fusion

We propose a hierarchical fusion, which fuses linguistic behaviour outputs (P_G) with content gate output (P_C) and target gate output (P_T) respectively and uses two CNNs to integrate that fusion to get C_C and C_T, then we concatenate C_C and C_T then flatten it to F_F. Finally, a multi-layer perceptron (MLP) is used for final prediction:

$$L_1 = ReLU(W_1 \cdot F_F + b_1), \quad L_2 = ReLU(W_2 \cdot L_1 + b_2), \quad Z = softmax(W_3 \cdot L_2 + b_3) \quad (5)$$

Three layers are stacked. For the each layer, W_i and b_i represent the weight matrix and bias vector, and the ReLU activation function is used for the first two layers. For the last layer, to get the probability of each class Z, softmax layer is used.

4 Evaluation Setting

We conducted experiments on all seven datasets with and without GCN as well as using the three different types of cross-transformer variances, which will be discussed in 5.2. The GCN embedding dimension for this linguistic behaviour graph is $D_{LBG} = 200$. For transformer encoder configuration, we used dropout rate = 0.5, encoder number = 2, head number = 3, and hidden dimension = 1296. The models are trained with batch size = 16, and lr(learning rate) and number of epochs differ: **Waseem**: lr = 4e-4, epochs = 6, **HatEval**: lr = 1e-7, epochs = 6, **OffEval**: lr = 1e-7, epochs = 13, **Davids**: lr = 4e-4, epochs = 6, **Founta**: lr = 1e-5, epochs = 8, **FNUC**: lr = 1e-6, epochs = 13, **StormW**: lr = 1e-6, epochs = 7. The hyper-parameters are decided by splitting the training set into 90:10 training and validation set.

The followings are the models evaluated in our experiments. **TF-IDF features and SVM Classifier (TIS)**: TIS (Yin et al., 2009) applies TF-IDF with SVM Classifier to detect abusive language. First, TF-IDF weights of words are generated and a Support Vector Machine with radial basis function (RBF) kernel is trained to classify different kinds of abusive languages. **One-Two Steps Hybrid CNN (OTH)**: OTH (Park and Fung, 2017) used a Hybrid CNN (word-level and character-level) model and applied it to abusive tweet detection. We applied Chars2vec as a character embedding and Glove as a word embedding. The convolutional layers with kernel size 256, 128, and 64 are stacked, and the model is trained using learning rate 4e-5 with 10 epochs. **Multi-Features with RNN (MFR)**: MFR (Mehdad and Tetreault, 2016) used a hybrid character-based and word-based Recurrent Neural Network (RNN) model to detect abusive language. After the Chars2vec and Glove embeddings, there is a vanilla stacked RNN. Three RNN layers with hidden dimensions 128, 128, and 64 are stacked, and the model is trained using learning rate 4e-6 with 10 epochs. **Two-step Word-level LSTM (TWL)**: TWL (Badjatiya et al., 2017) produced LSTM-derived representations with a Gradient Boosted Decision Trees classifier. The model applied LSTM to Glove embeddings, and the results are fed into the model. Three LSTM layers with hidden dimensions 128,128,64 are stacked, and the model is trained using learning rate 4e-6 with 10 epochs. **Latent Topic Clustering with Bi-GRU (LTC)**: LTC (Lee et al., 2018) applies a Bi-GRU with latent topic clustering, which extracts the topic information from the aggregated hidden states of the two directions of the Bi-GRU. Three Bi-GRU layers with hidden dimensions 128, 128, and 64 are stacked, and the model is trained using learning rate 4e-5 with 10 epochs. **Character-based Transformer (CBT)**: CBT (Bugueño and Mendoza, 2019) uses a transformer-based classifier with Chars2vec embeddings. Transformer encoders with hidden dimension 400, learning rate 4e-6 with 3 epochs are used.

5 Experiments and Results

5.1 Performance Comparison

In this part, we compare our model with six baseline models over all seven datasets, discussed in Sec 2.1. These baseline models are constructed with various word representations as well as different neural

Dataset/Algorithm	TIS	OTH	MFR	TWL	LTC	CBT	Ours
Waseem(Waseem and Hovy, 2016)	83.56	79.10	62.39	73.88	79.94	79.11	86.00
HatEval(Basile et al., 2019)	41.63	40.48	53.17	52.03	53.14	49.25	53.97
OffEval(Zampieri et al., 2019)	75.37	76.84	55.59	67.15	77.90	58.71	78.80
Davids(Davidson et al., 2017)	88.11	88.37	79.44	83.74	87.56	88.94	90.34
Founta(Founta et al., 2018)	79.58	78.59	73.64	75.23	79.49	72.04	80.36
FNUC(Gao and Huang, 2017)	68.92	64.51	70.71	65.67	69.78	67.07	73.20
StormW(de Gibert et al., 2018)	82.73	85.48	82.06	81.91	83.83	82.90	85.86

Table 2: Overall f1 results from seven ALD models (including *MACAS*) evaluated across all seven benchmark datasets. We highlight the top 2 models for each dataset, using darker colors for higher performance. For all the benchmark datasets, we train models on the train split and report results on test splits.

Dataset	Abusive Example	Actual C.	Correctly C.	Incorrectly C.
Waseem	Really **bitch** really? Get on the runway and **back in the kitchen**.	Sexism	TIS, Ours	OTH, MFR, TWL, LTC, CBT
Waseem	thegeek_**chick** Please answer...	Sexism	TIS, Ours	OTH, MFR, TWL, LTC, CBT
Waseem	RT randomfox: #ManSpreading **#FeminismIsAwful #WomenAgainstFeminism**	Sexism	TIS, Ours	OTH, MFR, TWL, LTC, CBT
Waseem	RT @asredasmyhair: **Fems, for the love of God**, please stop propagating the wage gap myth. #FemFreeFriday	Sexism	TIS, Ours	OTH, MFR, TWL, LTC, CBT
FNUC	**BLM is a Hate group** Given Birth by Obama's comments That put a target on the backs of police officers	Hateful	Ours	TIS, OTH, MFR, TWL, LTC, CBT
StormW	i am a **white nationalist** of a christian faith but still am a **white nationalist** for racial survival the anti racist christians are the true chirstians enemy	Hate	OTH, Ours	TIS, MFR, TWL, LTC, CBT

Table 3: Some prediction example from our model *MACAS* compared to the baselines. Actual C.: Actual Class, Correctly C.: Correctly Classified Models, Incorrectly C.: Incorrectly Classified Models.

networks or classifiers. Table 2 presents the weighted average f1 performance of each baseline model and our model over each dataset. Our model outperforms the baseline models for all these seven datasets. Applying multiple aspect embeddings enables our model to process the texts from multi-perspective views. The Cross-Attention gate flow makes it possible to obtain the mutual enhancement between the two different aspects. Although some of the baseline models such as OTH, MFR also combine two embedding approaches (Chars2vec and Glove) to get more information, they still just consider the general information of the texts rather than extract information in a targeted fashion from various aspects. For these reasons our model can achieve performance above the baseline models.

As well as comparing our model with the baseline models, we also make some observations from comparing the six baseline models amongst themselves. Firstly, OTH and MFR use the combined embeddings of Chars2vec and Glove which gives more information. So, they can achieve relatively better weighted average f1 scores compared to most other baseline models which just use a single embedding method. Secondly, the results of TWL and LTC indicate that the bi-directional recurrent neural network leads to better performance than the simple forward recurrent neural network. This means that not only the future states but also the past ones will affect the prediction results. Thirdly, although we may not consider TF-IDF with SVM to be as good as Chars2vec or Glove with deep neural networks, TIS baseline model never gets the worst weighted f1 score for the seven datasets when compared with other models. In fact it even outperforms other baseline models on **Waseem** and **Founta**. For both datasets, there might be some particular words which are really significant for identifying the class. So TF-IDF can achieve good results for these two datasets.

Next, an analysis of examples to show how different models perform on different datasets is conducted, and the results are shown in Table 3. The first example is from **Waseem**, "Really bitch really? Get on the runway and back in the kitchen.", which should be predicted as Sexism. It is quite explicit in that the word "bitch" is in this sentence, and this makes TIS predict it as Sexism easily since TF-IDF is focusing on the word occurrence. Besides, "back in the kitchen" is implicit Sexism, implying women should be in the kitchen. The similar patterns can be found from the second instance "thegeek_chick please answer" by explicitly mentioning the word 'chick'. The third and fourth samples represent abusive language or hate speech about the topic Feminism. The third explicitly stated the words 'Feminism' and 'Awful' and TIS and our model successfully detected the abuse with an explicit hate speech aspect identification. Our model, which considers the explicit and implicit aspects, can predict the sentence as Sexism easily. Another example is from **FNUC**, "BLM is a Hate group Given Birth by Obama's comments That put a

Methods	Waseem	HatEval	OffEval	Davids	Founta	FNUC	StormW
CB, no G	82.35	53.97	78.80	90.34	80.36	65.31	82.90
CB, G, N=1	86.00	51.88	75.06	87.36	76.90	73.20	84.52
CB, G, N=3	83.76	42.71	75.03	88.24	75.42	68.39	85.86
CBM, no G	81.53	53.28	77.37	90.25	80.28	65.67	84.14
CBM, G, N=1	85.22	39.91	72.60	90.12	76.06	68.92	85.09
CBM, G, N=3	82.77	42.86	75.10	90.11	77.03	68.16	85.12

Table 4: Abusive language detection results across seven benchmark datasets for *MACAS* with two cross attention aspect gate flow mechanisms and graph embedding. We highlight the top 2 settings for each dataset. The darker the colour, the better the performance. The comparison provides different parameters (*N*) of final fusion layers, including N=1 or 3. (CB: cross-attention at the beginning, CBM: cross-attention at the beginning and the middle, G: the user linguistic behaviour graph embedding)

target on the backs of police officers" which should be Hateful. This comment insults the "Black Life Matters" by calling it a Hate Group. Normally, describing something as a hate group is not hate speech, but in this case, calling BLM a hate group is racism. This is not easy for the baseline models to spot, and only our model predicts it correctly. For the last example from **StormW**, "i am a white nationalist of a christian faith but still am a white nationalist for racial survival the anti racist christians are the true chirstians enemy", the user described himself as "white nationalist" which is one kind of hate speech, and OTH can predict this sentence as Hate. The reason is that the CNN used in OTH can capture the information for phrases, which is the "white nationalist" here. Besides, our model can predict this sentence correctly since the sentence is a general explicit hate speech.

5.2 Ablation Testing - Cross-attention gate flow

In this part, three different structures of cross transformer encoders are tested: 1) Cross-transformer at the beginning of the the transformer encoder **(CB)**: exchanging content's and target's K and V at the beginning of the transformer encoders as in Figure 2; 2) Cross-transformer in the middle of the transformer encoder **(CM)**: exchanging content's and target's input for Feed Forward layer in the transformer encoder, which is in the middle of the transformer encoders; 3) Cross-transformer at both places **(CBM)**: the combination of CB and CM. Due to the poor performance of CM, only results for 7 datasets with CB and CBM structure are shown in Table 4. Besides, to find whether and how GCN is improving the performance of our model, different structures are also compared: 1) Model without GCN; 2) Model with GCN using hierarchical fusion, repeating one or three times. We show one and three times here because on all the datasets our model achieves the best performance with one or three repeated fusions when GCN is also used. Two conclusions are drawn based on the results of CB and CBM:

Firstly, the best model is always the CB model, and the second best is always the CBM model with the same GCN structure. So comparing between CB and CBM structure, CB has a better performance and we use this structure as our final model. Besides, in most cases, CB outperforms CBM if they share the same GCN structure, which also shows that, overall, CBM is worse than CB. Considering the fact that CM is the worst, we can say that cross in the middle transformer encoder will lower the model performance. Exchanging content's and target's K,V is important since it allows target aspects to query on the content aspects and vice versa. However, exchanging values before Feed Forward Layer only gives a different add and norm which doesn't increase the interaction between content aspects and target aspects usefully.

Secondly, our model can have a better performance with GCN when there is user id in the dataset. Not all the datasets provide user id, and as mentioned in Sec 3.1, User Linguistic Behavior embedding is trained by using the user id as the target. For those datasets without userid, the real abusive labels are used as the training target. By comparison, we can find that **Waseem**, **StormW**, and **FNUC** which provide user id in the datasets have a better performance using a model with GCN, and the other four datasets, which don't provide user id, have a better performance using a model without GCN. Therefore, for the dataset with user id, User Linguistic Behavior which is from GCN, can improve the performance of our model. And for those datasets without user id, the model structure without GCN is recommended.

Combinations	Waseem	HatEval	OffEval	Davids	Founta	FNUC	StormW
D + E	80.16	49.94	75.81	89.58	80.02	66.03	82.23
D + I	61.93	47.04	54.63	68.27	67.88	64.51	81.91
D + E + I	80.57	47.11	69.95	87.11	79.80	64.03	81.91
G + E	79.67	52.78	76.95	86.92	79.39	65.56	84.85
G + I	80.10	53.63	57.38	87.52	79.35	64.24	81.91
G + E + I	79.12	48.71	72.17	89.19	79.14	65.96	82.04
D + G + E	78.63	53.60	73.78	88.51	80.23	68.28	82.44
D + G + I	79.74	52.65	75.12	89.76	79.76	65.31	83.44
D + G + E + I	82.35	53.97	78.80	90.34	81.57	65.31	83.93

Table 5: Ablation studies comparing different types integration of multi-aspects for the generic ALD model. In the proposed model, *MACAS*, we introduced four aspect embeddings, including directed abuse (D), generalised abuse (G), explicit abuse (E), and implicit abuse (I). Directed and generalised abuses are in the group of a target aspect, while explicit and implicit abuses are in a content aspect group. The ablation testing is conducted in a different combination of aspect embedding from each higher-level of aspect groups. The highest performance is highlighted in green, the lowest is marked in red.

5.3 Ablation Testing - Multi-aspect embedding

To check how aspect embeddings contribute to the model, an ablation test on different combinations of the embeddings is conducted on all these seven datasets. We use the CB model without GCN for the prediction. Table 5 presents the weighted average f1 scores for 9 different combinations of four aspect embedding models, including Directed abuse D, Generalised abuse G, Explicit abuse E, and Implicit abuse I. Each target and content aspect should include at least one embedding.

For **Waseem**, the $D + G + E + I$ combination achieves the best performance with the weighted average f1 score 82.35 and most other combinations have a slightly lower performance. In contrast, $D + I$ gets the worst weighted f1 score of 61.93. The reason why $D + I$ is much worse than other combinations may lie in two facts: 1) In this dataset, abusive language is generally more explicit rather than directly aiming at a specific target in an implicit way. 2) Even humans can not distinguish Direct Abuse in an Implicit way easily, and it can be very difficult for the annotators to annotate the label correctly. Besides, the $D + G + E + I$ combination outperforms other cases because it takes all the aspects into consideration. Similar results occur on other Twitter datasets **Davids**, **HatEval**, **OffEval** and **Founta**, $D + G + E + I$ achieves the best while $D + I$ is much worse. For **FNUC**, due to the small volume of dataset and imbalanced labels, not all the combinations have a good prediction result. $D + G + E$ having the best performance implies that the dataset doesn't have a large number of implicit abuse samples. For **StormW**, $D + G + E + I$ gets the best performance. Besides, $G + E$ also has a good performance. The reason is that this dataset is collected from a racism forum and most hate speech on that website is generally abusive in an explicit way. Based on the analysis of the different embedding combinations on these datasets, we can conclude that the embeddings used may vary based on different kinds of datasets, but combining them all is always a good idea. Although four specific different embeddings are selected in our model to represent four different aspects, other kinds of embeddings could also be used as long as they can represent the corresponding aspects.

6 Conclusion

Abusive language detection is an essential but challenging task, and it is almost impossible to successfully encompass all different abusive language tasks in different domains. The evaluation also shows that most of the state-of-the-art ALD algorithms do not generalise their model to different types of abusive language problems or datasets. In this paper, we proposed a new generic abusive language model, called MACAS, which applied multi-aspect embeddings to represent generalised characteristics of the domain and introduced a cross-attention gate flow model to achieve better performance by mutual enhancement between the target aspect and the content aspect. The results indicate that our framework was successful and effective in capturing abusive language aspects in different domains. Compared to other ALD models, our model successfully works in general abusive language detection, and it is hoped that MACAS provides some insight into the future direction of generic abusive language detection.

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.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63.

Margarita Bugueño and Marcelo Mendoza. 2019. Learning to detect online harassment on twitter with the transformer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 298–306. Springer.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*.

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20.

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.

Antigoni-Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. In *11th International Conference on Web and Social Media, ICWSM 2018*. AAAI Press.

Lei Gao and Ruihong Huang. 2017. Detecting online hate speech using context aware models. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 260–266.

Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 161–169.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Masahiro Kaneko and Danushka Bollegala. 2019. Gender-preserving debiasing for pre-trained word embeddings. In *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Younghun Lee, Seunghyun Yoon, and Kyomin Jung. 2018. Comparative studies of detecting abusive language on twitter. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 101–106.

Yashar Mehdad and Joel Tetreault. 2016. Do characters abuse more than words? In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 299–303.

Cade Metz and Mike Issac. 2019. Facebook's a.i. whiz now faces the task of cleaning it up. sometimes that brings him to tears. *The New York Times*.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2018. Author profiling for abuse detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1088–1098.

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*, pages 145–153. International World Wide Web Conferences Steering Committee.

Ji Ho Park and Pascale Fung. 2017. One-step and two-step classification for abusive language detection on twitter. In *Proceedings of the First Workshop on Abusive Language Online*, pages 41–45.

Jing Qian, Mai ElSherief, Elizabeth Belding, and William Yang Wang. 2018. Leveraging intra-user and inter-user representation learning for automated hate speech detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 118–123.

Amir H Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. 2010. Offensive language detection using multi-level classification. In *Canadian Conference on Artificial Intelligence*, pages 16–27. Springer.

Ellen Spertus. 1997. Smokey: Automatic recognition of hostile messages. In *Aaai/iaai*, pages 1058–1065.

Julien Tissier, Christophe Gravier, and Amaury Habrard. 2017. Dict2vec : Learning word embeddings using lexical dictionaries. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Copenhagen, Denmark, September. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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.

Zeerak Waseem, 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.

Zeerak Waseem. 2016. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In *Proceedings of the first workshop on NLP and computational social science*, pages 138–142.

Jun-Ming Xu, Kwang-Sung Jun, Xiaojin Zhu, and Amy Bellmore. 2012. Learning from bullying traces in social media. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 656–666. Association for Computational Linguistics.

Dawei Yin, Zhenzhen Xue, Liangjie Hong, Brian D Davison, April Kontostathis, and Lynne Edwards. 2009. Detection of harassment on web 2.0. *Proceedings of the Content Analysis in the WEB*, 2:1–7.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, 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.