MC-BERT4HATE: Hate Speech Detection using Multi-channel BERT for Different Languages and Translations

Hajung Sohn
School of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology
Gwangju, Korea
hajungsohn@gist.ac.kr

Hyunju Lee
School of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology
Gwangju, Korea
hyunjulee@gist.ac.kr

***Abstract*—The growth of social networking services (SNS) has altered the way and scale of communication in cyberspace. However, the amount of online hate speech is increasing because of the anonymity and mobility such services provide. As manual hate speech detection by human annotators is both costly and time consuming, there are needs to develop an algorithm for automatic recognition. Transferring knowledge by fine-tuning a pre-trained language model has been shown to be effective for improving many downstream tasks in the field of natural language processing. The Bidirectional Encoder Representations from Transformers (BERT) is a language model that is pre-trained to learn deep bidirectional representations from a large corpus. In this paper, we propose a multi-channel model with three versions of BERT (MC-BERT), the English, Chinese, and multilingual BERTs for hate speech detection. We also explored the usage of translations as additional input by translating training and test sentences to the corresponding languages required for different BERT models. We used three datasets in non-English languages to compare our model with previous approaches including the 2019 SemEval HatEval Spanish dataset, 2018 GermEval shared task on the identification of Offensive Language dataset, and 2018 EvalIta HaSpeeDe Italian dataset. Finally, we were able to achieve the state-of-the-art or comparable performance on these datasets by conducting thorough experiments.**

***Index Terms*—BERT, Deep Learning, Hate speech Classification, Social Networking Services, Transfer Learning**

I. Introduction

The detection of hate speech in online posts and comments has become an issue of growing importance in recent years. Along with the growth of social networking services (SNS), hate speech in the cyberspace continues to increase because of the anonymity and mobility offered by social medias. Although there are various definitions of 'hate speech', the definition by Nockleby [1] is generally accepted in former literature.

*"**Hate speech** is defined as any communication that disparages a person or a group on the basis of some characteristic such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristic."*

Some examples [2] of hate speech in SNS include the following:

- *@USER nigga are you stupid your trash dont play with him play with your bitch*
- *#Conservatism101 It's not about our disagreements with #Conservatives. Its that Conservatives can't debate honestly, and they have no integrity. Whatever gets them thru today, is all that matters to them. They're fundamentally dishonest people. URL*

This type of language is considered as a social problem because most of the contents aim disadvantaged social groups and can further lead to the development of organized hate-based activities. With the increase in the social impact of hate speech over the past years, the interests of the research community, including governments, SNS companies, and individual researchers, in recognizing hate speech, have grown. Although many SNS companies have hired human annotators to manually filter out hate speech, they are still criticized for not doing enough [32]. As the manual detection of hate speech is both costly and time consuming, automatic detection methods are required.

There are some characteristics in hate speech that complicate its automatic identification without the involvement of human annotators. First, there is no absolute standard for what comprises a hate speech. The standard of offensive languages can vary based on country, time, culture, and political propensity. Thus, it is hard to merge several hate speech datasets that are annotated by different criteria in order to increase the data size. Second, hate speech cannot be identified by merely checking whether a swear word appears in a comment. For example, sarcasm is often used in hate speech and those phrases require a high-level understanding of context and nuance. Thus, the classifier for hate speech should be able to understand the multiple features to correctly identify the intention of the content.

Despite these difficulties, much research was conducted to resolve this issue. Hate speech detection has been usually considered as a binary classification task, but more fine-grained classification, such as predicting the hate speech type or the degree of aggressiveness, can also be included. Previous studies primarily considered the problem as a supervised sentence or document classification task [4]. Some used feature engineering and then fed the refined features to classifiers such as the support vector machine (SVM), Naive Bayes, and logistic regression. Others used the deep learning paradigm that employs deep neural network architectures to automatically learn hidden features. However, methods using transfer learning were not examined much in previous works.

Recently, transfer learning in deep learning algorithms was shown to outperform existing methods for many natural language processing (NLP) tasks [5]–[7]. The Bidirectional Encoder Representations from Transformers (BERT) is an unsupervised language model that was trained on a very large corpus and can be used to transfer knowledge to different downstream tasks through fine-tuning [8]. Therefore, by extracting features or fine-tuning BERT, we can successfully transfer knowledge learned from language models to hate speech classifiers.

In addition, in sentence classification tasks, the use of additional context related sentences may improve the classification performance. Utilizing translated sentences as auxiliary inputs in sentence classifiers is another way to use related sentences. This method was not considered much in the past because it was generally assumed that translations have more noise than meaningful signals. The idea was first suggested by Amplayo *et al.* [9] who showed that when the translated data was modified properly to reduce noises, they could be effectively applied in several sentence classification tasks. We suggest another way to reduce the noise in translations. The error in translations can be remedied by fine-tuning pre-trained language models because pre-trained models may be capable of extracting meaningful information and neglect errors in translation results since they are trained on a very large formal corpus.

Our objective in this study is to develop a multi-channel BERT model for different languages to apply transfer learning in hate speech detection. We also investigate the effect of using translations as supplementary inputs. We demonstrate that our model and mechanisms outperform or have similar performance with existing state-of-the-art approaches when considering three hate speech datasets. We tested the model on different datasets in different languages to show that our method is easily applicable to hate speech detection in different languages.

II. Related works

In this study, we use ‘hate speech’ as a general term for numerous kinds of insulting expressions or statements, covering similar concepts such as ‘toxic sentence’, ‘offensive language’, ‘trolling’, and ‘cyberbullying’. Unlike other terms, ‘cyberbullying’ especially refers to hate speech targeting an individual rather than a group. The methods that researchers used can be divided into three categories: conventional machine learning, deep learning methods, transfer learning using BERT. Conventional machine learning methods rely on manual feature engineering that is then consumed by supervised machine learning algorithms such as SVM and logistic regression. Deep learning methods represent deep learning mechanism that uses neural networks. Transfer learning using BERT refers to more recent transfer learning methods that use pre-trained language models.

A. Conventional Machine Learning Methods

For conventional machine learning methods, feature selection is one of the most important standard for grouping different approaches. [4] Surface features, such as bag-of-words, bag-of-characters, character n-grams, and word n-grams are used as meaningful features to predict hate speech [10]. Mehdad and Treault [11] reported that character-based features were more contributive than word based features to hate speech detection because character n-grams are useful for attenuating the spelling variation and new word problem often faced when working with internet comments.

Word- and character- based features are generally combined with other linguistic, syntactic, and distributional features to boost performance. Linguistic features such as the number of words in a comment, mention of a URL, and the number of punctuations was also used to solve the problem of noisiness in comments by Nobata *et al.* [12]. Alfina *et al.* [13] used character n-grams, word n-grams, the number of negative sentiment words and classifiers such as SVM, random forest, naive bayes, and logistic regression on a self-constructed Indian Twitter hate speech dataset. Syntactic information such as part-of-speech (POS) information was applied along with ngrams by Xu *et al.* [14]. Chen *et al.* [15] employed dependency relation such as the relationship between the target and offensive terms, as another syntactic feature. Distributed word representations learned from neural networks, have also been used as important features. For example, GloVe is a word embedding trained to effectively capture the distributional information between words [16]. Fasttext is a word embedding that include subword information [17]. By using such word vectors, similar sentences with the same words usually have a similar hidden representation. Atalaya *et al.* [18] used a combination of bag-of-words (BoW), bag-of-characters (BoC), and fasttext embeddings as sentence representation before feeding it to an SVM classifier. Lexicons such as the offensive word dictionary were utilized along with other core features by Chen *et al.* [15] and Dadvar *et al.* [20]. In addition, Dadvar *et al.* [19] used features outside text such as user characteristic and profile information, as improving factors for predicting hate speech.

Among the classifiers, the SVM was the most used algorithm for classification [32], whereas, Naive Bayes, logistic regression, and random forest were also used frequently.

B. Deep Learning Methods

Deep neural networks improved performance of many NLP problems. For sentence classification, neural networks including the Convolutional Neural Network (CNN) [21], Recurrent Neural Network (RNN) [22], Long-Short-Term-Memory (LSTM) model [23], and Gated Recurrent Unit (GRU) [24] have been used to boost performance. For word-level deep learning models, word embeddings such as GloVe [16], Fasttext [17], and word2vec [25] can be used as input representations. Mehdad and Tetreault [11] used the RNN for hate speech classification. The CNN architecture was employed by several researchers in competitions over shared tasks [26]–[29], [32]. The LSTM model was used by Pitsilis *et al.* [30]. Nikhil *et al.* [31] used the LSTM with self-attention model to concentrate on more important sequential units. A CNN-GRU model was proposed to classify tweets into categories of racism, sexism, and neither of them by Zhang *et al.* [32]. A Bidirectional LSTM-CNN model was proposed by Wiedemann et al. [33]. Grunigen *et al.* [34] suggested a parallel CNN model with the GRU model.

C. Transfer Learning from BERT

Transfer learning is a way to enhance learning in a new task by transferring knowledge from a related task [35]. In the field of NLP, unsupervised language model pre-training has been proved to be effective for improving downstream tasks. **BERT**, which stands for Bidirectional Encoder Representations from Transformers, is a language model that is pre-trained with a very large corpus to learn deep bidirectional representations [8]. It uses multiple layers of the bidirectional transformer, which is a powerful model for sequence-to-sequence tasks [36], to be trained jointly for two tasks. The first task is the masked language model which aims to predict the masked word that is randomly selected from the original text. An example of a masked sentence is as follows:

Input Text: the man jumped up , put his basket on phil ##am ##mon ' s head

Masked Input: [MASK] man [MASK] up , put his [MASK] on phil [MASK] ##mon ' s head

The other task is the next sentence prediction task, which predicts whether a sentence pair is consecutive. The following is an example of a continuous sentence pair:

Sentence A: the man went to the store .

Sentence B: he bought a gallon of milk .

These two tasks are pre-trained simultaneously. After the pre-training is complete, there are two ways to transfer knowledge from the pre-trained BERT to downstream tasks: feature-based approach and fine-tuning. The feature-based approach involves the extraction of hidden sentence representation from the BERT model, while fine-tuning adds an additional task-specific layer above the BERT model, and then trains all parameters together. Fine-tuning BERT for sentence classification tasks has proved to be effective in supervised sentence classification tasks including hate speech detection [2].

III. Proposed Methodologies

A. Dataset description

The HatEval dataset is a Spanish train and test dataset for the 2019 SemEval task 5, of multilingual detection of hate speech against immigrants and women in Twitter [37]. The comments were collected from the Spanish Twitter and labeled by experts and non-trained annotators from a crowdsourcing platform.

The GermEval dataset comprises data on the 2018 shared task on the identification of offensive language [38]. The collected sentences were labeled as not-hate or hate by one of the task organizers.

The HaSpeeDe dataset comprises data on the 2018 EvalIta task, HaSpeeDe [39]. The Twitter corpus was developed by the Turin group [40] in 2007. After the first annotation step that resulted in a collection of approximately 1,800 tweets, the corpus was further expanded by additional annotations. The added data were annotated by experts and a crowdsourcing platform. The label distribution of all three datasets is depicted in Figure 1.

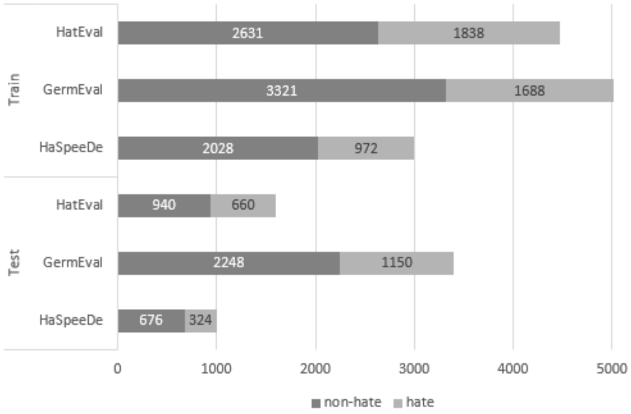


Fig. 1. Distirbution of the three datasets: HatEval, GermEval, HaSpeeDe.

B. Dataset preprocessing

Posts and comments in social media frequently use hashtags, emojis, user mentions, and URLs. Thus, it is important to conduct proper processing prior to training. For fast and reliable preprocessing, we used the open source ekphrasis library(https://github.com/cbaziotis/ekphrasis), which is a preprocessing tool specialized for Twitter. The following are the steps of our procedure:

- The URL in each tweet was replaced by the <url> tag and the user mention was replaced by the <user> tag.
- Then, elongated words such as 'yaaaaay' were normalized to be 'yaay'.
- Punctuations and special characters were removed to make the text cleaner, while hashtags and emojis were not removed and used to be embedded.

The overall preprocessing procedure is summarized in Figure 2

C. Using translations

We also applied translations to make parallel pseudo data in other languages. In sentence classification tasks, the use of additional context related sentences may improve the performance. Amplayo *et al.* [9] suggested translations are valuable input features if properly modified to extract signals and remove errors. He pointed out using additional contexts such as neighboring sentences can give positive effect on

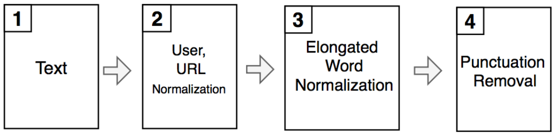


Fig. 2. Data preprocessing flow

classifiers, but such features are domain-dependent and cannot be applied with single sentences without any context. However, translations are domain-free contents and available without the domain. Therefore, utilizing translated sentences as auxiliary inputs in sentence classifiers is another way to improve the accuracy of sentence classifiers regardless of the domain. The translated text can be used as inputs to a single language specific fine-tuning BERT model. Otherwise, it could be used as a supplementary input for multi-channel BERT (MC-BERT) that unites three different BERT models for different languages. We used the Google Translation API to translate text of source language to English and Chinese text to feed the English and Chinese BERT. The translation process is depicted in Figure 3. We will show using BERT, the model could be more robust to errors when using translations.

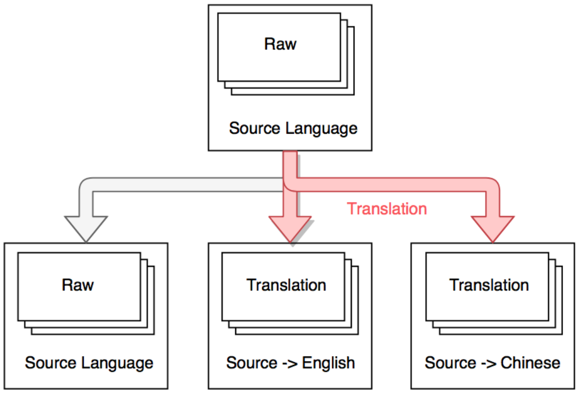


Fig. 3. translation generation with Google translation API

D. *Baseline: BERT fine-tuning*

We used the default input representation of BERT for fine-tuning of sentence classification.

The right side of each sentence is padded with [PAD] tokens or truncated to match equal sequence length. Also, a [CLS] token is appended to the left of all sentences since it was used for learning the whole sentence representation during the pre-training phase. BERT input is composed of three features: byte pair encodings (BPE) embedding, segment embedding, and positional embedding.

$$\mathbf{BERTInput}(x) = \mathbf{BPEE}(x) + \mathbf{SE}(x) + \mathbf{PE}(x) \tag{1}$$

BERT first splits an input sentence x to BPE tokens. BPE [41] is a mean of splitting sentences into units smaller than words to handle out of vocabulary words more effectively. Segment embedding is the same for tokens that are not [PAD] tokens. Positional embedding is projected as follows, where i is the index of the token and j is the index of the dimension. $model_d$ is the dimension size which all three embeddings are the same.

$$PE(i, 2j) = \sin(\frac{i}{10000^{2j/model_d}}) \tag{2}$$

$$PE(i, 2j+1) = \cos(\frac{i}{10000^{2j/model_d}}) \tag{3}$$

The pre-trained BERT has two parameter intensive settings.

BERT$_{\text{BASE}}$(uncased, English): 12 layers, 768 hidden dimensions, 12 attention heads (in transformer), total number of 110M parameters.

BERT$_{\text{LARGE}}$(uncased, English): 24 layers, 1024 hidden dimensions, 16 attention heads (in transformer), total number of 340M parameters.

We used the pre-trained **BERT**$_{\text{BASE}}$ model for the English BERT. In addition to the English version, the Chinese and multilingual BERTs were also trained and released with the same parameter as those of **BERT**$_{\text{BASE}}$ but with different vocabulary files and language of the training corpus. As there are currently three BERT models available, there are three possible fine-tuning models if parallel language data are provided. We used the multilingual BERT for the original language dataset which were in non-English language, and used parallel translations for the English and Chinese BERT fine-tuning.

The fine-tuning procedure is simple. We first initialize the model weights by using the pre-trained language model parameters. Then, from the last layer of BERT, the hidden representation of the [CLS] token is pooled by the pooling layer. A dropout layer is then used for regularization. Finally, a fully connected feed-forward layer and a softmax layer is used for classification.

$$\mathbf{o} = \mathbf{W}\mathbf{x} + \mathbf{b} \tag{4}$$

$$P(c|x,\theta) = \frac{exp(o_c)}{\sum\limits_{c\in\{hate,not-hate\}} exp(o_c)} \tag{5}$$

The feed forward layer is a single layer that multiplies a weight matrix $\mathbf{W}$ and adds bias $\mathbf{b}$ to the pooled output $\mathbf{x}$. The final output of the model should be $P(c|x)$, which is the probability for a sentence x to be classified as class c. After the feed forward layer predicts o_c for each class, a softmax operation is done for normalizing the output to be between zero and one.

The total fine-tuning model is depicted in Figure 4.

E. *Multi-channel BERT architecture*

This section explains the main multi-channel BERT (MC-BERT) model for different languages we propose, as shown

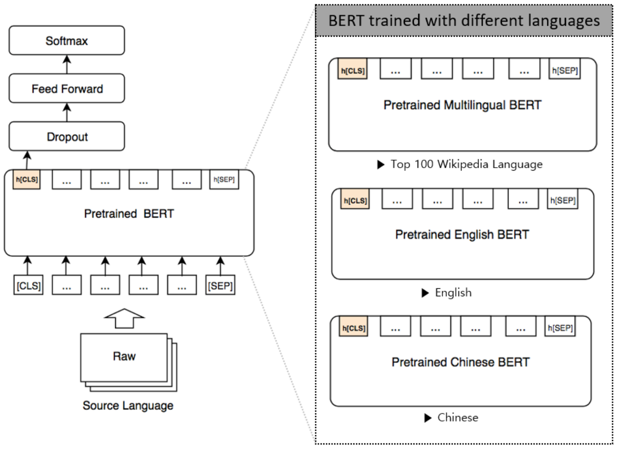


Fig. 4. Baseline model: Fine-tuning BERT for different languages

in Figure 5. We appended an adding layer after all single fine-tuning models to make a joint representation of three BERT models. Each hidden state was added as a weighted sum.

$$\mathbf{h}_{MC} = w_1 \cdot \mathbf{h}_{multilingual} + w_2 \cdot \mathbf{h}_{english} + w_3 \cdot \mathbf{h}_{chinese} \quad (6)$$

$\mathbf{h}_{MC}$ is the weighted sum of the hidden state of different BERT models, which indicates the hidden state of the multi-channel model. w_i is the weight multiplied to the hidden states, $\mathbf{h}_{multilingual}$, $\mathbf{h}_{english}$, and $\mathbf{h}_{chinese}$. All the weights are greater than zero and smaller than one.

The weights were manually set proportional to the fine-tuning performance. For example, if the Chinese fine-tuning model has a low accuracy than other BERT fine-tuning models, the multi-channel model provides some weight to the pooling output of the Chinese model. After the hidden states were added, the result was fed to another fully connected feed forward layer and softmax layer for the final classification.

F. Experimental Setup

When training the model, the batch size was 32. Adam [44] was used as the optimizer with a learning rate of 2e-5. The basic objective function of the model is the cross entropy loss.

$$J_\theta(y', y) = -\frac{1}{N}\sum_i (y'_i \log(y_i) + (1 - y'_i)\log(1 - y_i)) + \lambda||\theta||^2 \quad (7)$$

It would be $y'_i = 1$ if the true class of the sentence is the ith label, and $y'_i = 0$ otherwise. y_i is the probability of the sentence to be classified to the ith label. L2 regularization with weight decay and dropout [45] were also used to prevent overfitting and make the model more robust to noises.

The training epoch for each dataset was different. While the HatEval dataset was trained for 4 epochs, the GermEval and HaSpeeDe datasets were trained for 2 epochs. The hyper-parameters used are summarized in Table I.

TABLE I
HYPER-PARAMETERS

Batch size	32
Learning rate	2e-5
Optimizer	Adam
L2 weight decay	0.01
Dropout rate	0.1
Loss function	Categorical cross entropy

IV. RESULTS

A. Metrics

The results of the models were evaluated based on two metrics: the accuracy and F_1 macro score. The F_1 macro score is calculated by plain averaging of the F_1 score of all classes. To provide a measure that weights all labels equally, macro-F_1 score was chosen to evaluate classifiers for hate speech among all F1 averaging schemes. The metrics can be computed as follows:

$$Accuracy = \frac{number\ of\ correctly\ predicted\ instances}{total\ number\ of\ instances}$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

$$F_1 - Macro = \frac{F_1(not - hate) + F_1(hate)}{2}$$

B. HatEval Result

The best fine-tuning model among different BERT models was the multi-channel BERT fine-tuning model, which scored the highest accuracy and F1 score, of 0.769 and 0.766, respectively. The accuracy was 3.8% higher than that obtained from the state-of-the-art model. The previous state-of-the-art model used rich featured Spanish GloVe vectors with the SVM classifier. Each sentence vector was considered as a combination of bag-of-words (BoW), bag-of-characters (BoC), and fasttext embeddings [18]. The English and multilingual BERT showed similar performance to the multi-channel BERT. The fine-tuning accuracy of the English BERT was 0.752 and F1 score was 0.748. The accuracy and F1 score of the multilingual BERT fine-tuning were 0.755 and 0.751, respectively. The Chinese BERT showed comparatively low performance to other BERT models with an accuracy of 0.700 and F1 score of 0.690. Except for the Chinese BERT model, all fine-tuning models outperformed the previous state-of-the-art model results. This shows that transferring knowledge makes the model more robust to translation errors. The experimental results are summarized in Table II. The two dimensional PCA result of the hidden representation before the final feed forward layer for English BERT, multilingual BERT, and multi-channel BERT is depicted in Figure 6. As shown in Figure 6 (A), (B), and (C), each BERT fine-tuning model captures different the hidden representations. This shows that even though the model is similar, different languages can learn different features for classification.

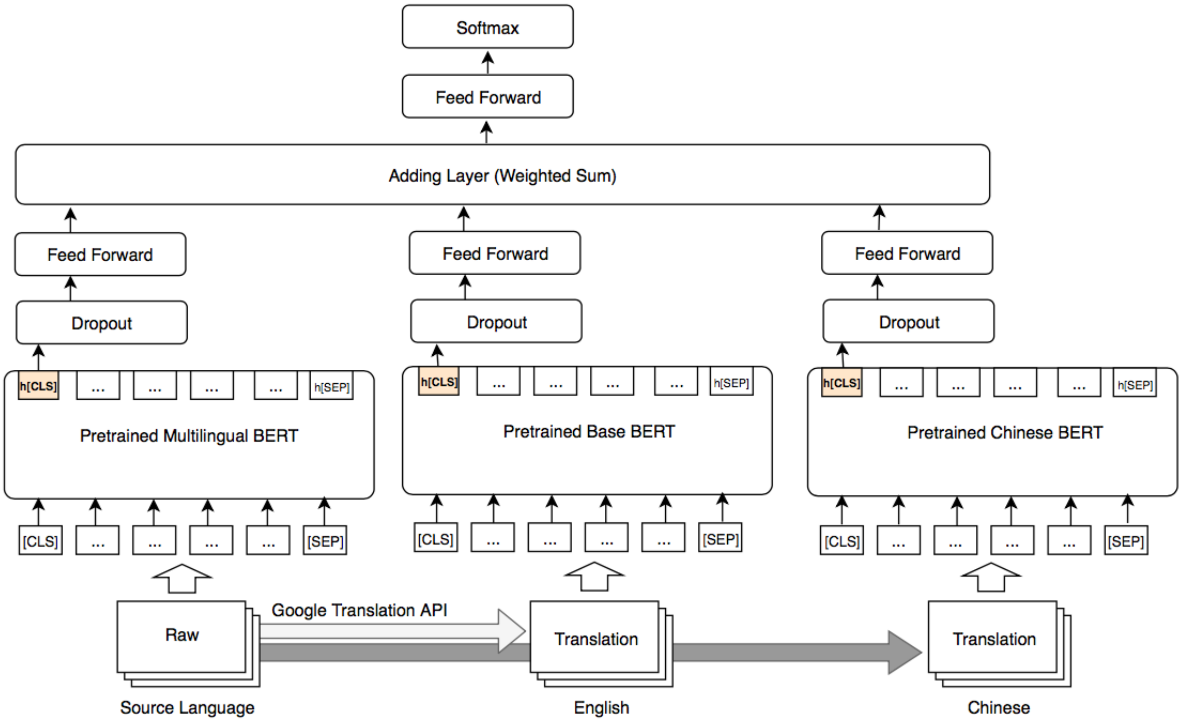


Fig. 5. Multi-channel BERT for different languages (MC-BERT)

C. GermEval Result

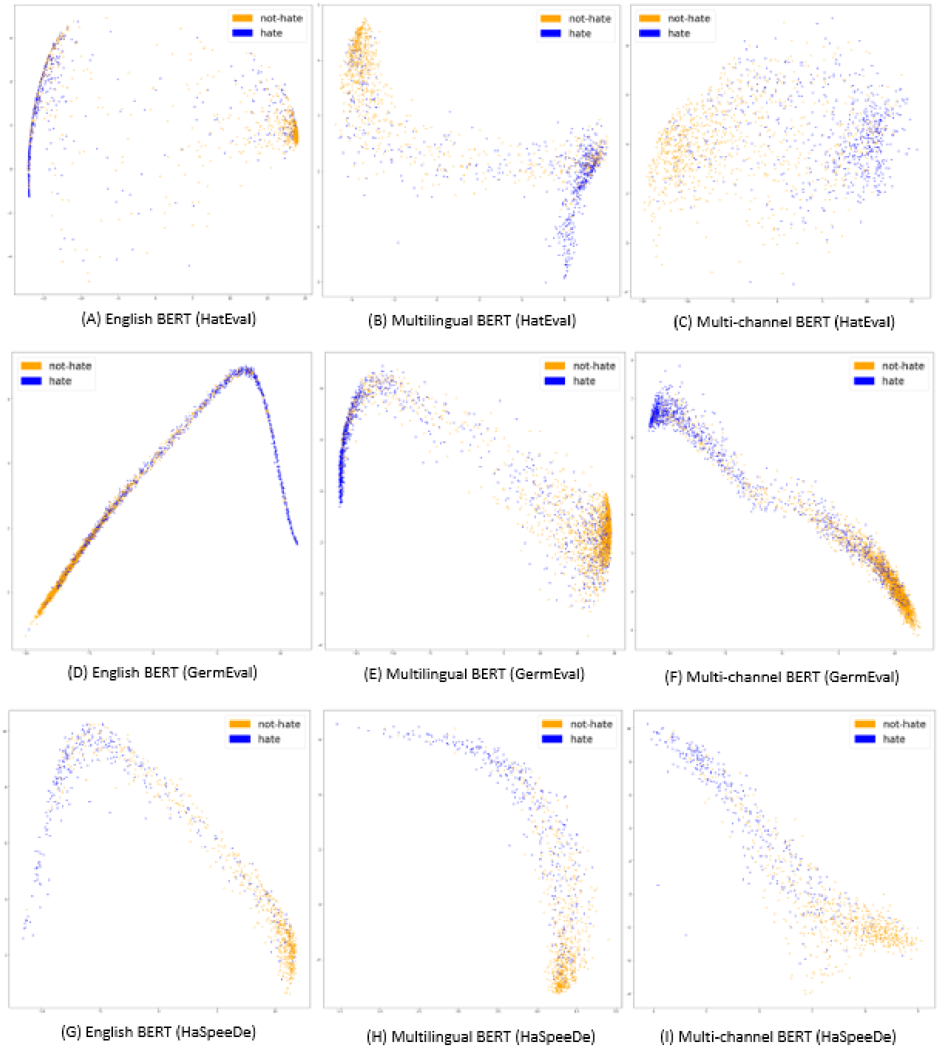
The comparison of the different BERT fine-tuning models showed that the English and the multi-channel BERTs recorded the highest F1 score and fine-tuning accuracy. While the English BERT fine-tuning showed an accuracy of 0.798 and F1 score of 0.770, the multi-channel BERT fine-tuning showed an accuracy of 0.801 and F1 score of 0.764. The accuracy and the F1 score of the multilingual BERT were 0.779 and 0.742, which are respectively 3.0% lower accuracy and 3.8% lower than those of the multi-channel and English BERT model. The Chinese BERT fine-tuning showed a higher accuracy than that of the LSTM model using fasttext embeddings, which was 0.760. However, the Chinese model still performed poorly than the other fine-tuning models possibly because of the translation errors. The best model outperformed the state-of-the-art model, which was an ensemble model of logistic regression and random forests using word n-gram, character n-gram, and word embeddings [42]. The previous state-of-the-art performance on the test set showed accuracy and F1 score of 0.795 and 0.767, respectively. The results of our approach are shown in Table II. Figure 6 depicts the visualization of the two dimensional PCA of the hidden state before the final classification layer of different BERT models. You can see from Figure 6 (D), (E), and (F) that different features are extracted from different fine-tuning models.

D. HaSpeeDe Result

The best fine-tuning performance was that of the multilingual BERT model, which showed fine-tuning accuracy of 0.822 and F1 score of 0.799. The other English, Chinese, and multi-channel BERT showed similar performance. The English BERT showed fine-tuning accuracy and F1 score of 0.798 and 0.773, respectively, while these values for the Chinese BERT were 0.799 and 0.775. The multi-channel model showed an accuracy and F1 score of 0.800 and 0.775, respectively, which are the highest among all the BERT fine-tuning models, except the multilingual BERT. Our best scoring model showed a higher accuracy than that of the state-of-the-art model, which was an ensemble model of SVM and bidirectional LSTM using additional data. Although the accuracy is not reported, the F1 score was 0.799 [43]. Even though our model did not outperform the previous state-of-the-art result, it showed comparable performance. Table II summarizes our results of our approaches on the HaSpeeDe test dataset. The visualization of the two dimensional PCA of the hidden state before the classification layer of different BERT fine-tuning models is depicted in Figure 6.

V. Conclusion

This study was aimed at developing an effective model for automatic hate speech detection. The proposed model integrates the hidden features of separate BERT models trained on different languages. This approach is supposed to effectively capture different semantic representation of different languages. In addition, we investigated the effect of translations as auxiliary sentences for sentence classification. We tested our methods on three datasets from different competitions in different languages. In all datasets, the multi-channel BERT fine-



(A) English BERT (HatEval)

(B) Multilingual BERT (HatEval)

(C) Multi-channel BERT (HatEval)

(D) English BERT (GermEval)

(E) Multilingual BERT (GermEval)

(F) Multi-channel BERT (GermEval)

(G) English BERT (HaSpeeDe)

(H) Multilingual BERT (HaSpeeDe)

(I) Multi-channel BERT (HaSpeeDe)

Fig. 6. PCA visualization of the hidden representation before the final feed forward layer in the HatEval, GermEval, HaSpeeDe dataset

tuning model or baseline model with translations exceeded or performed as well as the previous state-of-the-art models. The recent contribution of transfer learning in NLP is impressive. Owing to the use of transfer learning, the problem of shortage in labeled datasets can be remedied by pre-training a language model on a very large corpus. Our research also showed that transfer learning is effective for hate speech detection. Moreover, we demonstrated that although translations from machine translation models have many errors, they are positive supplementary inputs for text classification. We expect that our experiments contribute toward the further study of text mining in social media and knowledge transfer.

References

[1] J. T. Nockleby, "Hate speech," in Encyclopedia of the American Consitution, 2nd ed., K. L. K. Leonard W. Levy and D. J. Mahoney, Eds. Macmillan, 2000, pp. 1277-1279.

[2] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval)," 03 2019.

[3] Z. Zhang, D. Robinson, and J. Tepper, "Detecting hate speech on twitter using a convolution-gru based deep neural network," in The Semantic Web, A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam, Eds. Cham: Springer International Publishing, 2018, pp. 745-760.

[4] A. Schmidt and M. Wiegand, "A survey on hate speech detection usingnatural language processing," inProceedings of the Fifth InternationalWorkshop on Natural Language Processing for Social Media. Valencia,Spain: Association for Computational Linguistics, Apr. 2017, pp. 1-10.

[5] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, andQ. V. Le, "Xlnet: Generalized autoregressive pretraining for languageunderstanding,"CoRR, vol. abs/1906.08237, 2019

[6] A. Radford, "Improving language understanding by generative pretraining," 2018.

[7] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in Proc. of NAACL, 2018.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pretraining 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). Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171-4186.

TABLE II
HatEval, GermEval, HaSpeeDe Test Set Result

HatEval Results

Method	Accuracy	F1 Macro
SVC Baseline [37]	0.705	0.701
Glove + LSTM	0.716	0.710
(SOTA) BoW+ BoC + fasttext + SVM (Perez and Luque, 2019) [18]	0.731	0.730
English BERT fine-tune	0.752	0.748
Multilingual BERT fine-tune	0.755	0.751
Chinese BERT fine-tune	0.700	0.690
Multi-channel BERT fine-tune	**0.769**	**0.766**

GermEval Results

Method	Accuracy	F1 Macro
Fasttext + LSTM	0.700	0.638
(SOTA) ngram + ensemble of LR and RF (Montani and Schuller, 2018) [42]	0.795	0.767
English BERT fine-tune	0.798	**0.770**
Multilingual BERT fine-tune	0.771	0.732
Chinese BERT fine-tune	0.760	0.720
Multi-channel BERT fine-tune	**0.801**	0.764

HaSpeeDe Results

Method	Accuracy	F1 Macro
Fasttext + LSTM	0.783	0.753
Fasttext + SVM	-	0.774
(SOTA) SVM + (bi)LSTM + additional data (Cimino et al., 2018) [43]	-	**0.799**
English BERT fine-tune	0.798	0.773
Multilingual BERT fine-tune	**0.822**	**0.799**
Chinese BERT fine-tune	0.799	0.775
Multi-channel BERT fine-tune	0.800	0.775

[9] R. K. Amplayo, K. Lee, J. Yeo, and S. won Hwang, “Translations as additional contexts for sentence classification,” in IJCAI, 2018. [80] M. Sanguinetti, F. Poletto, C. Bosco, V. Patti, and M. Stranisci, ”An Italian twitter corpus of hate speech against immigrants,” in Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018). Miyazaki, Japan: European Languages Resources Association (ELRA), May 2018.

[10] T. Davidson, D. Warmsley, M. W. Macy, and I. Weber, “Automated hate speech detection and the problem of offensive language,” CoRR, vol. abs/1703.04009, 2017.

[11] Y. Mehdad and J. R. Tetreault, “Do characters abuse more than words?” in SIGDIAL Conference, 2016.

[12] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, “Abusive language detection in online user content,” in Proceedings of the 25th International Conference on World Wide Web, ser. WWW 16. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2016, pp. 145-153.

[13] I. Alfina, R. Mulia, M. I. Fanany, and Y. Ekanata, “Hate speech detection in the indonesian language: A dataset and preliminary study,” in 2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Oct 2017, pp. 233-238.

[14] J.-M. Xu, K.-S. Jun, X. Zhu, and A. Bellmore, “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, ser. NAACL HLT ’12. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 656-666.

[15] Y. Chen, Y. Zhou, S. Zhu, and H. Xu, “Detecting offensive language in social media to protect adolescent online safety,” in Proceedings of the 2012 ASE/IEEE International Conference on Social Computing and 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust, ser. SOCIALCOM-PASSAT 12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 71-80. [Online]. Available: http://dx.doi.org/10.1109/SocialCom- PASSAT.2012.55

[16] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532-1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[17] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” CoRR, vol. abs/1607.04606, 2016. [Online]. Available: http://arxiv.org/abs/1607.04606

[18] J. M. Pérez and F. M. Luque,“Atalaya at SemEval 2019 task 5: Robust embeddings for tweet classification,” in Proceedings of the 13th International Workshop on Semantic Evaluation. Minneapolis, Minnesota, USA: Association for Computational Linguistics, Jun. 2019, pp. 64-69.

[19] M. Dadvar, D. Trieschnigg, R. Ordelman, and F. de Jong, “Improving cyberbullying detection with user context,” in Advances in Information Retrieval, P. Serdyukov, P. Braslavski, S. O. Kuznetsov, J. Kamps, S. Ruger, E. Agichtein, I. Segalovich, and E. Yilmaz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 693-696.

[20] M. Dadvar, F. de Jong, R. Ordelman, and D. Trieschnigg, “Improved cyberbullying detection using gender information,” 01 2012.

[21] Y. Kim, “Convolutional neural networks for sentence classification,”arXivpreprint arXiv:1408.5882, 2014.

[22] J. L. Elman, “Finding structure in time,”Cognitive science, vol. 14, no. 2,pp. 179-211, 1990.

[23] S. Hochreiter and J. Schmidhuber, “Long short-term memory,”Neural com-putation, vol. 9, no. 8, pp. 1735-1780, 1997.

[24] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, “Empirical evaluationof gated recurrent neural networks on sequence modeling,”CoRR, vol.abs/1412.3555, 2014

[25] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in Advances in neural information processing systems, 2013, pp. 3111- 3119.

[26] A. Ribeiro and N. Silva, “INF-HatEval at SemEval-2019 task 5: Convolutional neural networks for hate speech detection against women and immigrants on twitter,” in Proceedings of the 13th International Workshop on Semantic Evaluation. Minneapolis, Minnesota, USA: Association for Computational Linguistics, Jun. 2019, pp. 420-425.

[27] A. Husseini Orabi, M. Husseini Orabi, Q. Huang, D. Inkpen, and D. Van Bruwaene, “Cyber-aggression detection using cross segment-and-concatenate multi-task learning from text,” in Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018). Santa Fe, New Mexico, USA: Association for Com- putational Linguistics, Aug. 2018, pp. 159-165.

[28] A. Rozental and D. Biton, “Amobee at semeval-2019 tasks 5 and 6: Multiple choice cnn over contextual embedding,” in SemEval@NAACL-HLT, 2019.

[29] J. G. R. de Sousa, “Feature extraction and selection for automatic hate speech detection on twitter,” 2019.

[30] G. K. Pitsilis, H. Ramampiaro, and H. Langseth, “Detecting offensive language in tweets using deep learning,” arXiv preprint arXiv:1801.04433, 2018.

[31] N. Nikhil, R. Pahwa, M. K. Nirala, and R. Khilnani, “Lstms with attention for aggression detection,” in TRAC@COLING 2018, 2018.

[32] Z. Zhang, D. Robinson, and J. Tepper, “Detecting hate speech on twitter using a convolution-gru based deep neural network,” in The Semantic Web, A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam, Eds. Cham: Springer International Publishing, 2018, pp. 745-760.

[33] G. Wiedemann, E. Ruppert, R. Jindal, and C. Biemann, “Transfer learning from lda to bilstm-cnn for offensive language detection in twitter,” ArXiv, vol. abs/1811.02906, 2018.

[34] D. von Grünigen, R. Grubenmann, F. Benites, P. Von Däniken, and M. Cieliebak, “spmmmp at germeval 2018 shared task: Classification of offensive content in tweets using convolutional neural networks and gated recurrent units,” 09 2018.

[35] E. S. Olivas, J. D. M. Guerrero, M. M. Sober, J. R. M. Benedito, and A. J. S. Lopez, Handbook Of Research On Machine Learning Applications and Trends: Algorithms, Methods and Techniques - 2 Volumes. Hershey, PA: Information Science Reference - Imprint of: IGI Publishing, 2009.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” inAdvances in neuralinformation processing systems, 2017, pp. 5998-6008.

[37] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. Rangel, P. Rosso, and M. Sanguinetti, “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 (SemEval-2019).

Association for Computational Linguistics", location = "Minneapolis, Minnesota, 2019.

[38] M. Wiegand, M. Siegel, and J. Ruppenhofer, "Overview of the germeval 2018 shared task on the identification of offensive language," 09 2018.

[39] C. Bosco, D. Felice, F. Poletto, M. Sanguinetti, and T. Maurizio, "Overview of the evalita 2018 hate speech detection task," in EVALITA 2018-Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, vol. 2263. CEUR, 2018, pp. 1-9.

[40] M. Sanguinetti, F. Poletto, C. Bosco, V. Patti, and M. Stranisci, "An italiantwitter corpus of hate speech against immigrants, inProceedings of the EleventhInternational Conference on Language Resources and Evaluation (LREC-2018).

[41] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715-1725. [Online]. Available: https://www.aclweb.org/anthology/P16-1162

[42] J. P. Montani, "Tuwienkbs at germeval 2018: German abusive tweet detection," in 14th Conference on Natural Language Processing KONVENS 2018, 2018, p. 45.

[43] A. Cimino, L. D. Mattei, and F. Dell'Orletta, "Multi-task learningin deep neural networks at EVALITA 2018," inProceedings of theSixth Evaluation Campaign of Natural Language Processing and SpeechTools for Italian. Final Workshop (EVALITA 2018) co-located withthe Fifth Italian Conference on Computational Linguistics (CLiC-it2018), Turin, Italy, December 12-13, 2018., 2018.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization,"CoRR,vol. abs/1412.6980, 2014.

[45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov,"Dropout: a simple way to prevent neural networks from overfitting,"The jour-nal of machine learning research, vol. 15, no. 1, pp. 1929-1958, 2014.