MGEL: Multi-Grained Text Representation Analysis and Ensemble Learning in Online Abusive Language Detection

Anonymous EMNLP submission

Abstract

In this work, we describe our efforts in fighting against abusive language and present insights gained. Specifically, we conduct a comprehensive multi-grained text representation analysis on current popular language models from crude word segmentation to single-byte granularity. We have found that granularity significantly impacts the empirical performance of the model, to the extent that a simple linear model could also beat well-tuned CNN and BiLSTM although more compact multi-hot byte-level quantization and subword schemes are introduced to boost them. As a continual effort towards the fight against abusive language, we introduce an enhanced BERT, on which subword works well for context understanding but performs poorly on intentional obfuscations. We propose to rescue its deficiency by integrating byte and character and develop a Multi-Grained Ensemble Learning (MGEL) framework. It advances the state-of-the-art performance on the largest abusive language datasets as demonstrated by our evaluation.

1 Introduction

It is notoriously risky for online audiences to be exposed to abusive language when they engage on social media, which could have a negative impact on the integrity of online communities. Thus, there have been continued efforts cracking down on toxicity from different media platforms including setting up standards and guidelines for potential users, human moderation, and machine learning detection systems (Nobata et al., 2016; Badjatiya et al., 2017; Schmidt and Wiegand, 2017). The profound impacts of toxic contents can extend from cyberspace to the physical security of enterprise and even the entire society. For instance, the allegations against social media, especially Facebook, with regard to Russia's 2016 election-meddling has forced the company to overhaul the News Feed and hire additional moderators[1]. In some cases, machine learning-based moderation systems could also mark ordinary contents as abusive language mistakenly[2].

Therefore, it has been important and challenging to understand and develop models to to detect toxic user generated contents with high accuracy. Previous studies have undertaken pioneering explorations on this topic. Most works treat toxic comments detection in the same way generic text classification is carried out or alternatively focus on certain ethnic groups or building up blacklists of swear words (Yin et al., 2009; Warner and Hirschberg, 2012; Sood et al., 2012; Nobata et al., 2016; Badjatiya et al., 2017). The involved features above are all limited to words or character levels.

Perpetrators often intentionally obfuscate certain words about groups, or abusive words, by misspelling, or leetspeak (e.g., "/\/1gger", "ph*ck", "w.e.t.b.a.c.k.") (Perea et al., 2008), which could easily create new words not seen by a word-based model (Gröndahl et al., 2018). To alleviate this, a slightly finer granularity of subwords can be leveraged to better capture word obfuscation, as well as the out-of-vocabulary problem (Wu et al., 2016; Devlin et al., 2018). On the other hand, character-level features are demonstrated better than word-level ones in text classification (Zhang et al., 2015; Kim et al., 2016), especially for processing less curated user-generated texts. The downside, however, is that it only works well on the single-byte character set. When it comes to the multi-byte characters (e.g., CJK and Emojis), vocabulary has to be large enough to cover them, which could be problem-

[1] https://www.vanityfair.com/news/2018/08/facebooks-hate-speech-problem-may-be-bigger-than-it-realized

[2] https://abcnews.go.com/beta-story-container/US/facebook-blocks-restores-declaration-independence-post/story?id=56383239

atic for the one-hot encoding scheme. To address this, we introduce more fine-grained byte-level decomposition into abusive language study, which provides a more compact representation.

In the domain of abusive language detection, the state-of-the-art performance (SOTA) come from Bidirectional LSTM (BiLSTM) and attention based Bidirectional Encoder Representations from Transformers (BERT) (Agrawal and Awekar, 2018; Bodapati et al., 2019). However, the systematic studies on text representation remain absent. To this end, we investigate how word, subword, character and byte shape their performance on large-scale datasets totaling over 4 million examples (the largest one so far). More importantly, although classical machine learning methods are not shown competitive in existing studies, we revisit them and introduce a simple yet effective algorithm. In addition, we propose an enhanced BERT architecture that outperforms the SOTA. Finally, we do ensemble learning by integrating classical machine learning and enhanced BERT for further advancing the state-of-the-art performance of abusive language detection.

The main contributions are: (1) pushing the state-of-the-art performance on the largest and comprehensive abusive datasets so far; (2) performing the first systematic study exploring multi-grained text representation including byte for abusive text and offering useful insights.

2 Related Work

Early studies on the toxicity detection took advantage of handcrafted generic features such as N-gram, Term Frequency-Inverse Document Frequency (TF-IDF), regular expressions patterns, lexical, parser, linguistic, syntactic, semantic and contextual features to distinguish between the toxic comments and ordinary ones (Warner and Hirschberg, 2012; Chen et al., 2012; Nobata et al., 2016). There are also specific studies devoted to certain ethnic groups (Warner and Hirschberg, 2012) and blacklist/swear words (Agrawal and Awekar, 2018), where large-scale labeling and annotation are generally performed by the crowd-sourcing Amazon Mechanical Turk workers and human moderators (Nobata et al., 2016). Recently, a growing number of end-to-end learning algorithms have also been proposed to fight against hate speech (Badjatiya et al., 2017; Gambäck and Sikdar, 2017; Zhang et al., 2018; Founta et al., 2019; Agrawal and Awekar, 2018; Bodapati et al., 2019). BiLSTM and BERT achieve the best performance in the race (Agrawal and Awekar, 2018; Bodapati et al., 2019).

To better model the irregularity of natural language such as rare and unknown words, researchers has explored various granularities of text modeling: word embedding enriched with sub-word information (FastText) (Bojanowski et al., 2017), byte-pair encoding (BPE) (Sennrich et al., 2016), byte-pair embedding (BPEmb) or sub-word embedding ("wordpieces") (Wu et al., 2016), and character based learning (Zhang et al., 2015; Kim et al., 2016) have been proposed. Byte-level inputs were also explored in other fields like Named Entity Recognition and Part-of-Speech tagging with multilingual backgrounds (Irie et al., 2017; Gillick et al., 2015). Recently, there is a good study on word decomposition models for abusive language detection (Bodapati et al., 2019). Our work, however, differs from all the above approaches in several aspects. First, we focus on abusive language and offer comprehensive studies including a proposed byte quantization scheme. Another one reason is that multi-byte characters are in tiny proportion. We also explore the potential of classical machine learning models besides recently developed ones, which turns out to be a very strong candidate.

3 Datasets

We prepare four datasets including audiences' reactions (comments) to Yahoo! Finance and Yahoo! News, Wikipedia talk pages and Twitter (Agrawal and Awekar, 2018) toxicity and hate speech datasets.

Table 1: Basic statistics of data including irregular text (column % Ir.) and abusive in irregular text (% Ab. in Ir.). posts with at least one multi-byte character and those with only single-byte characters are referred to be *irregular* and *regular*, respectively

Source	# Abusive	# Clean	Total	% Abusive	% Ir.	% Ab. in Ir.
Finance	34,839	1,072,724	1,107,563	3.1%	4.7%	3.9%
News	177,419	2,635,179	2,812,598	6.3%	3.0%	5.2%
Wikipedia	13,590	102,274	115,864	11.7%	7.9%	4.3%
Twitter	5,054	11,036	16,090	31.4%	10.2%	37.9%

Finance and News sets are sampled comments posted for articles in Yahoo! Finance and Yahoo! News between January 24, 2013 and January 23, 2018, spanning 1825 days. Original comments are roughly grouped into abusive and clean categories, respectively. Abusive comments are annotated out of toxic categories. This broadly includes hate speech, profanity, derogatory language, etc. Further details on the collection and labeling of these data-sets can be found in (Nobata et al., 2016). Duplication is applied to remove redundancy. The breakdown of clean and abusive comments is reported in Table 1.

Wikipedia and Twitter datasets are more focused on cyberbullying languages, which include abusive languages that belong to any of the following categories: personal attack, sexism, and racism. Specifically, the corpus of Wikipedia and Twitter have about 116K labeled discussion comments and 6K annotated tweets, respectively (Agrawal and Awekar, 2018). We group all posts into abusive and clean according to whether they are cyberbullying languages, which are detailed in Table 1 as well. In our subsequent experiments, we use 80% of the data for training models (60% for training and 40% for development) and perform model evaluation on the remaining 20% as the test data.

We list the details of the above datasets in Table 1 and compute various statistics measures on the abusive level. To specifically distinguish posts with at least one multi-byte character and those with only single-byte characters (referred to as *irregular* and *regular*, respectively), their statistics are derived separately.

4 Methodology

We describe the proposed algorithms, elaborate on the existing text representation and the proposed scheme.

4.1 Algorithms

4.1.1 NBLR

Although existing studies show that deep learning based models (e.g., CNN, LSTM) outperform traditional machine learning algorithms (e.g., logistic regression, random forest) in abusive language detection tasks (Agrawal and Awekar, 2018), we still believe that a carefully constructed linear method has a place in the tool chest of hatespeech detection, because such a method has good interpretability. In addition it is easy and fast to train, and stable and efficient to serve.

Bag of n-gram tokens and TF-IDF have been widely used for text classification (Nobata et al., 2016; Agrawal and Awekar, 2018). The odds ratio analysis[3] shows that prior count ratio of tokens between different classes is a reasonable metric to weight how well they are indicative of abuse. Thus, we propose to integrate them together and develop a variant based on logistic regression (LR) using Naive Bayes (NB) log-count ratios as feature weights (Wang and Manning, 2012). We call this algorithm NBLR for brevity, which is detailed in Algorithm 1.

Algorithm 1: Naive Bayes Logistic Regression (NBLR)

Input : Text corpus and labels (M samples)
Output : Logistic regression model
(1) Form word and character n-gram vector of N elements from the text corpus
(2) Compute the $M \times N$ TF-IDF matrix (sparse)
(3) Compute the log Naive Bayes ratio r_j for each column j of X , then scale the column with it
(4) Train a logistic regression using the scaled feature matrix X and the labels

The Naive Bayes ratio r_j of feature j measures the log-odds of the feature being associated with positive labels. More specifically, let the feature matrix be $X \in R^{M \times N}$, binary labels be $y \in \{0, 1\}^M$, then r_j is defined as the logarithm of the ratio between the average value of the elements of column j of X that are associated with positive labels, and the average value associated with negative labels.

4.1.2 Enhanced BERT

BERT has been widely demonstrated effective in multiple natural language processing tasks.

In this work, we propose an Enhanced BERT by making three-fold changes in comparison to (Bodapati et al., 2019): (1) adding the whole-word level positional embedding on top of the original overall one as shown in Fig. 1 (a); (2) masking bi-gram whole words instead of individual tokens as illustrated in Fig. 1 (b); (3) pre-training the model from scratch rather than only doing fine-tuning on small-scale datasets.

The added whole-word level positional embedding definitely introduces the extra complexity. The parameter complexity of BERT is given as $(V + F + S) \times H + L \times 12H^2 + H^2$, where V, F, S are vocabulary size, the maximum sequence length, segment type size. H and L is the hidden layer dimension and the number of transformer

[3] https://en.wikipedia.org/wiki/Odds_ratio

block layers, respectively. For Base size of BERT, $O \approx$ 110M. The added parameter number is absolutely less than $F \times H = 512 * 768 \approx 0.4$M, which is less than 0.4% increment. In addition, the new masking scheme doesn't introduce additional parameters. Thus, they work quite well in practice.

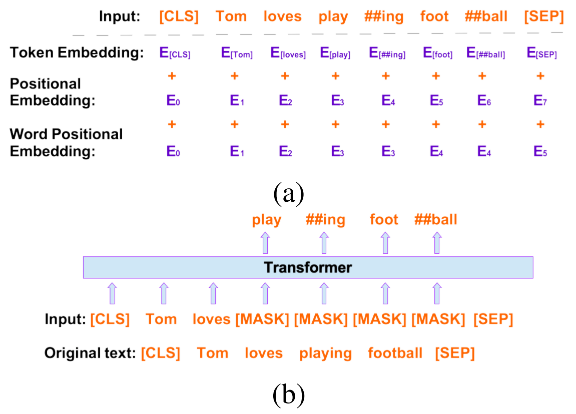

Figure 1: Enhanced BERT with (a) word-based positional embedding (b) bi-gram whole-word masking. Next sentence prediction task is removed as well. This is a subword model for illustration purpose, which can be readily applied to character and byte ones.

4.2 Text Representation

The textual inputs are typically decomposed into different granularities spanning from word (Mikolov et al., 2015), subword (wordpiece) (Wu et al., 2016), and character levels (Zhang et al., 2015; Kim et al., 2016) for the downstream learning in online abusive language detection (Bodapati et al., 2019). The byte-level decomposition, however, hasn't been explored in abusive language detection, albeit being studied elsewhere in different manner from our study as mentioned in section 2.

Word is the most frequently used textual decomposition unit. There are two main types that are of interest to our work: n-gram and word2vec embedding (Mikolov et al., 2013; Pennington et al., 2014). **Subword** is usually referred to as wordpiece (Wu et al., 2016), which could be implemented by the deterministic byte-pair encoding (BPE) (Sennrich et al., 2016) or probabilistic unigram language model (Kudo, 2018). It helps to alleviate the open vocabulary problems in different NLP tasks. In this work, we utilize Google's SentencePiece with unigram language model to generate subword vocabulary[4]. Given the generated subword vocabulary, we reformat text corpus and train subword embeddings based on word2vec from scratch. **Character** is the basic unit of text (Zhang et al., 2015). We here mainly utilize characters through one-hot encoding as described in (Zhang et al., 2015) and n-gram. The downside of the former is that alphabet size cannot be large enough to capture other non-English characters (e.g., CJK) due to the curse of dimensionality. Fortunately, non-English characters are often in tiny proportion. For character n-gram in linear model and vocabulary used in BERT models, character alphabets are not limited to the above.

In addition to word, subword and character, we propose to decompose text into **bytes** as well. Specifically, we encode all observed characters in the training data to obtain their corresponding UTF-8 codes[5] to generate a set of all unique bytes as the vocabulary for the byte-level quantization. Impressively, 206 bytes are sufficient to cover all characters for data sets used in this work. Given a character c, we retrieve its UTF-8 code denoted as $B = [b_1, ..., b_n]$, where $n \in \{1, 2, 3, 4\}$ corresponds to the encoding width. We then develop a multi-hot byte-level quantization scheme, as shown in Fig. 2. Then, each character is transformed to one m-sized vector, where an element corresponds to the count of the involved bytes. For instance, sequence *KDD* ☺Ω has 6 characters including a white space, which is denoted as 6×9 matrix $[[1,0,0,0,0,0,0,0,0]^T, [0,1,0,0,0,0,0,0,0]^T, [0,1,0,0,0,0,0,0,0]^T, [0,0,0,0,0,0,0,0,1]^T, [0,0,1,1,1,1,0,0,0]^T, \quad [0,0,0,0,0,0,1,1,0]^T]$ given a byte vocabulary {'0x4b', '0x44', '0xf4', '0x8f', '0xb0', '0x82', '0xce', '0xa9', '0x20'}. Vocabulary is built from UTF-8 codes for characters *K (0x4b)*, *D (0x44)*, *white space (0x20)*, *☺(0xf40x8f0xb00x82)* and *Ω (0xce0xa9)*. If all characters are with single byte ($n = 1$), the multi-hot byte-level scheme is equivalent to the character-level quantization (Zhang et al., 2015). We here don't report one-hot byte-level results due to its inferior performance.

4.2.1 MGEL

NBLR and Enhanced BERT are two totally different modeling schemes. The former one emphasizes the neighbor-to-neighbor interaction locally using traditional n-grams, whereas the latter one offers the deep all-to-all attention globally thorough modern transformers. On the other hand, the input of

[4] https://github.com/google/sentencepiece

[5] https://docs.python.org/3/howto/unicode.html

1 Byte	2 Bytes	3 Bytes	4 Bytes
1	0	0	0
0	1	0	1
0	1	2	1
0	0	0	2
0	0	1	0

Figure 2: Illustration of multi-hot (n-of-m) quantization scheme for characters with n (n=1,2,3,4) bytes. The number of rows is vocabulary size m.

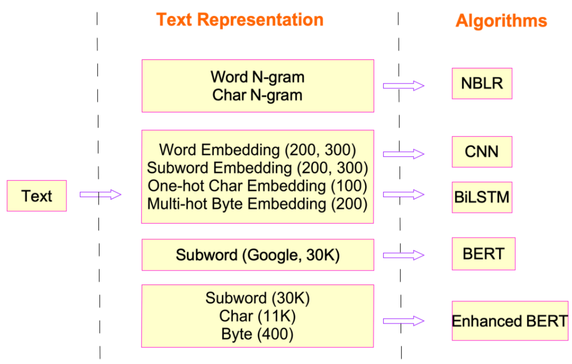

Figure 3: Diagram of text representations and algorithms employed. For word and subword embedding, the embedding dimensions are 200, 300 for Twitter and Wikipedia in Glove (Pennington et al., 2014). The word2vec embedding trained from scratch is set to have dimension of 300 (Mikolov et al., 2013). Original BERT has subword vocabulary of around 30K tokens from Google. Subword (eg., foot, ##ball), character and byte vocabularies are generated from abusive text corpora with sizes of 30K, 11K and 400, respectively.

NBLR can be thought of as the integration of word, subword, and character. Likewise, we propose to integrate byte, character and subword to form a Multi-Grained Enhanced BERT. We then further do ensemble learning of NBLR and Multi-Grained Enhanced BERT using a non-trainable simple algebraic combiner. Specifically, we have

$$u_j(x) = \sum_{c=1}^{C} w_c h_{c,j}(x) \tag{1}$$

where w_c is the weight assigned to the c^{th} classifier h. They can be obtained based on the validation performance. We call it Multi-Grained Ensemble Learning (MGEL).

5 Experiments and Results

In this section, we present the current state-of-the-art methods, evaluation metrics, experiment settings and a series of experiments to study text representations empirically. The whole study is summarized graphically in Fig. 3.

5.1 Baselines

Even though many algorithms have been developed for abusive language detection, the current state-of-the-art algorithms are Bidirectional LSTM (BiLSTM) and attention based BERT (Agrawal and Awekar, 2018; Bodapati et al., 2019). In addition, we include Convolutional Neural Networks (CNN) into baselines for the completeness.

CNN has also been effective in natural language processing recently (Zhang et al., 2015; Gambäck and Sikdar, 2017; Zhang et al., 2018; Irie et al., 2017). Character-level CNN (Char-CNN) has shown superior performance in text classification compared to other levels of representation (Zhang et al., 2015; Kim et al., 2016). In this context, we leverage classical temporal CNN (one-dimensional) as workhorse to perform textual representation and model comparison analysis.

BiLSTM and Gated Recurrent Unit (GRU) are both a recurrent neural network architecture, often used in sequence data modeling. Bi-GRU is on par with BiLSTM, thus we mainly study the granularity comparison on BiLSTM (Chung et al., 2014).

BERT is a recently developed self-supervised language model based on the Transformer encoder network (Devlin et al., 2018). Instead of ingesting the context from left to right or right to left in a sequential way as in recurrent neural network architecture (e.g., LSTM, BiLSTM), BERT proposes to enable tokens to have visibility of all other tokens. It has been employed in the fight against abusive language and demonstrates the state-of-the-art performance amongst a plethora of deep learning based advanced models with additional feature engineering (Bodapati et al., 2019).

5.2 Evaluation Metrics

To assess the detection capacity of different input granularities and algorithms, we adopt two metrics, namely, Area under Curves of Receiver Operating Characteristic (AUC@ROC) and Curves of Precision-Recall (AUC@PR) (Davis and Goadrich, 2006), respectively. In addition, we examine F1 score and Matthews correlation coefficient (MCC $\in [-1, 1]$, 1 for perfect prediction). These two metrics are based on a specific operating point and we take 0.5 in the involved experiments. MCC is generally regarded as a balanced measure.

5.3 Experiment Settings

We experiment NBLR with different combinations of word and character n-grams. It's found that word-level 1,2-gram and character-level 1,2,3,4-gram perform well in general. For temporal CNN and BiLSTM, we experiment with word-, subword-, character-, byte-level inputs, respectively. For word and subword embedding of Yahoo! Finance and News datasets, we utilize Gensim[6] on abusive text corpus to train embedding with output dimension of 300 and vocabulary size of 376K. For Wikipedia and Twitter, pre-trained Glove (Pennington et al., 2014) is used for embedding with the maximum corresponding output dimensions of 300 for Wikipedia and 200 for Twitter[7]. Character and byte vocabularies are generated from corresponding datasets. BERT has pre-trained models developed on standard text corpus including Wikipedia, which can be used for fine-tuning. Following (Bodapati et al., 2019), we take the uncased BERT-Base model as the starting point. The maximum sequence length is set to 300 for Finance, News and Wikipedia, 50 for Twitter as same as subwords in CNN and BiLSTM. We then fine-tune the model for respective datasets.

To make parameter tuning practicable, we set up the following rules for CNN and BiLSTM: (1) For word and subword, we tune hyper-parameters for the former and then apply them to the later. The rationale is that both textual decompositions generate similar distributions of textual length for same data sets. Likewise, we perform the hyper-parameter tuning for character and apply them to byte. Pre-trained embedding is utilized for feeding of word-level and subword-level inputs into models. For character and byte level inputs, one-hot and multi-hot representation is fed directly into the end-to-end learning as mentioned in preceding sections. (2) with regard to the datasets, Finance, News and Wikipedia share a common set of hyper-parameters. On the other hand, since Twitter is different from others in terms of both textual length and its distribution patterns, we have another set of hyper-parameters.

In this manner of fixing hyper-parameters, we attempt to make sure as much as possible that the performance discrepancy can be attributed to the difference in the textual decomposition approaches.

[6]https://github.com/RaRe-Technologies/gensim

[7]https://nlp.stanford.edu/projects/glove/ Wikipedia:glove.6B.zip, Twitter:glove.twitter.27B.zip

Table 2: Performance comparison among different decomposition approaches of textual input for CNN, BiLSTM and NBLR. Some results are based on multiple independent runs with mean and square bracketed standard deviation

Method	Source	Textual Input	AUC@ROC	AUC@PR	MCC	F1 Score
CNN	Finance	Word	0.8424[0.0032]	0.2284[0.0125]	0.1451[0.0641]	0.0902[0.0489]
		Subword	0.8862[0.0058]	0.3269[0.0164]	0.2774[0.0301]	0.1437[0.0453]
		Char	0.9089[0.0008]	0.4132[0.0047]	0.3696[0.0231]	0.3468[0.0423]
		Byte	**0.9128**[0.0013]	**0.4256**[0.0025]	**0.3815**[0.0172]	**0.3614**[0.0312]
	News	Word	0.8660[0.0020]	0.4786[0.0067]	0.4328[0.0111]	0.4143[0.0184]
		Subword	0.9078[0.0019]	0.5883[0.0057]	0.5277[0.0122]	0.5262[0.0216]
		Char	0.9277[0.0026]	0.6550[0.0087]	0.5928[0.0100]	0.6041[0.0128]
		Byte	**0.9301**[0.0007]	**0.6634**[0.0028]	**0.6017**[0.0066]	**0.6156**[0.0110]
	Wikipedia	Word	**0.9526**[0.0028]	**0.8202**[0.0081]	**0.7058**[0.0105]	**0.7304**[0.0130]
		Subword	0.9512[0.0032]	0.8144[0.0043]	0.6999[0.0055]	0.7227[0.0090]
		Char	0.9461[0.0023]	0.8067[0.0063]	0.6911[0.0065]	0.7153[0.0105]
		Byte	0.9483[0.0014]	0.8138[0.0034]	0.6964[0.0034]	0.7210[0.0065]
	Twitter	Word	0.8247[0.0083]	0.7174[0.0110]	0.5041[0.0121]	0.6264[0.0180]
		Subword	0.8401[0.0027]	0.7332[0.0051]	0.5096[0.0079]	0.6493[0.0102]
		Char	0.8465[0.0067]	0.7458[0.0083]	0.5348[0.0164]	0.6516[0.0189]
		Byte	**0.8568**[0.0125]	**0.7600**[0.0168]	**0.5487**[0.0207]	**0.6678**[0.0238]
BiLSTM	Finance	Word	0.8639[0.0009]	0.2801[0.0156]	0.2383[0.0611]	0.1927[0.0904]
		Subword	**0.8998**[0.0115]	0.3921[0.0134]	**0.3765**[0.0130]	**0.3637**[0.0333]
		Char	0.8834[0.0032]	0.3897[0.0041]	0.3363[0.0847]	0.2954[0.0038]
		Byte	0.8923[0.0027]	**0.3985**[0.0085]	0.3502[0.0057]	0.3020[0.0216]
	News	Word	0.8852[0.0036]	0.5214[0.0057]	0.4657[0.0129]	0.4522[0.0371]
		Subword	0.9262[0.0022]	0.6356[0.0033]	0.5708[0.0080]	0.5771[0.0204]
		Char	0.9250[0.0003]	0.6534[0.0016]	0.5977[0.0006]	0.6112[0.0001]
		Byte	**0.9269**[0.0010]	**0.6588**[0.0026]	**0.6012**[0.0037]	**0.6137**[0.0054]
	Wikipedia	Word	**0.9631**[0.0003]	**0.8450**[0.0047]	**0.7315**[0.0061]	**0.7598**[0.0013]
		Subword	0.9608[0.0022]	0.8359[0.0040]	0.7234[0.0059]	0.7512[0.0089]
		Char	0.9360[0.0001]	0.7870[0.0007]	0.6665[0.0260]	0.6988[0.0166]
		Byte	0.9352[0.0010]	0.7904[0.0042]	0.6773[0.0098]	0.7032[0.0021]
	Twitter	Word	0.8429[0.0007]	0.7423[0.0052]	0.5142[0.0149]	0.6362[0.0270]
		Subword	**0.8624**[0.0023]	**0.7636**[0.0042]	**0.5403**[0.0036]	**0.6758**[0.0152]
		Char	0.8328[0.0012]	0.7402[0.0085]	0.5152[0.0128]	0.6195[0.0209]
		Byte	0.8493[0.0080]	0.7580[0.0069]	0.5495[0.0105]	0.6595[0.0183]
NBLR	Finance	N-grams (Word, Char)	0.9388	0.4893	0.4028	0.3648
	News		0.9501	0.7149	0.6208	0.6206
	Wikipedia		0.9687	0.8674	0.7389	0.7533
	Twitter		0.9105	0.8454	0.6280	0.7116

5.4 Results

Tables 2 and 3 compare different representations for CNN, BiLSTM, NBLR and BERT models.

5.4.1 CNN

Overall, fine-grained approaches outperform coarse-grained ones clearly. Among all of them, byte-level representation achieves best performance across different datasets. The performance discrepancy stems from that the former can capture rare, unknown words, misspelling and morphology more effectively than the latter. This finding is in line with the related studies as well (Zhang et al., 2015; Gillick et al., 2015). The performance gain is also found to differ among different datasets. Specifically, the superiority of byte-level inputs is more evident in Finance and Twitter than that in News and Wikipedia. To untangle this point, we categorize all comments and tweets into two groups based on whether an online post has multi-byte characters. The single-byte set generally consists of limited ASCII characters, which can be fully captured by character-level quantization. The multi-byte character set has a large variety of characters. A much large number of input features are required to quantize them, which is not always feasible. As shown in Table 1, the overall percentages of abusive posts, irregular posts and percentage of abusive in irregular posts as reported in Table 1.

It's observed that the abusive percentage in irregular posts in Finance (3.9%) and Twitter (37.9%) is higher than the overall abusive percentage (3.1% and 31.4%). The difference of abusive percentage is completely reversed in both News (5.2% vs. 6.3%) and Wikipedia (4.3% vs. 11.7%). The higher percentage shows stronger signals of irregular text in indicating abusive language. This actually facilitates the advantages of using byte-level inputs, which can model irregular text smoothly.

For Wikipedia, neither character-level nor byte-level inputs outperform word-level and subword-level ones. In addition, the performance comparison between word-level and subword-level is reversed as well. This discrepancy might result from the difference of users in Wikipedia dataset in comparison to others. In Finance, News and Twitter datasets, general audience can post comments and tweets without needing much domain knowledge. Wikipedia itself is a collaborative knowledge repository. The dataset includes discussion among users who participated in its editing, which has some personal online attacks. Attacks are likely to be caused by disputes on specific domain knowledge. In this context, language styles are probably different from general posts in other media platforms. The percentage of abusive in irregular posts is almost one third of overall abusive percentage. In other words, the irregular characters are not good indicators of abusive language. Thus, the advantages of fine-grained inputs for capturing rare, unknown words are no longer beneficial.

5.4.2 BiLSTM

Similarly to CNN, byte and subword models perform better than character and word ones, respectively. Word and subword models are improved for BiLSTM in comparison to CNN as well. Both byte and character models, however, experience the performance deterioration to different extents, which leads to the reversal as we observe in Table 2. The underlying possibility is that LSTM cannot handle long sequence properly. For byte and character, input length has to be much longer than word and subword to cover input sequences. It is expected to get performance degraded due to gradient vanishing and exploding issues. Previous studies show that byte-level LSTM is the best one with only length of 60 (Gillick et al., 2015). In this work, the input sequence is usually a few hundreds.

Table 3: Performance comparison among different decomposition approaches of textual input with NBLR, BERT (Bodapati et al., 2019) (SOTA) and enhanced BERT (pre-training 20 epochs and fine-tuning)

Method	Source	Textual Input	AUC@ROC	AUC@PR	MCC	F1 Score
BERT	Finance	Subword	0.9490	0.5133	0.4687	0.4702
	News		0.9553	0.7276	0.6525	0.6685
	Wikipedia		0.9782	0.8932	0.7837	0.8063
	Twitter		0.9157	0.8483	0.6739	0.7726
Enhanced BERT	Finance	Subword	**0.9556**	**0.5484**	**0.5217**	**0.5369**
		Char	0.9479	0.5179	0.4908	0.5014
		Byte	0.9498	0.5129	0.4908	0.5028
	News	Subword	**0.9559**	**0.7300**	0.6451	0.6547
		Char	0.9519	0.7206	0.6512	0.6662
		Byte	0.9529	0.7237	**0.6543**	**0.6712**
	Wikipedia	Subword	**0.9814**	**0.9024**	**0.7933**	**0.8151**
		Char	0.9765	0.8897	0.7764	0.8022
		Byte	0.9775	0.8919	0.7862	0.8104
	Twitter	Subword	**0.9335**	**0.8746**	**0.6901**	**0.7811**
		Char	0.9180	0.8511	0.6721	0.7695
		Byte	0.9251	0.8656	0.6869	0.7800

5.4.3 NBLR and BERT

It's found that linear model NBLR works much better than advanced models CNN and BiLSTM as shown in Table 2. The fine-tuned BERT (Bodapati et al., 2019) models developed on standard text corpus achieves the best performance among the above algorithms as shown in Tables 2 and 3. It further demonstrates that the generic language modeling with reasonable training paradigm could even benefit abusive language detection greatly.

We also pre-train enhanced BERT models with subword-, character- and byte-level inputs on abusive text corpus. We can see that enhanced BERT with subword input is better than BERT model. The performance gains are much obvious in Twitter compared to others, which is related to the highest proportion of irregular text and abusive in them as shown in Table 1. It's also observed that byte is superior to character, which is consistent with the previous findings.

Table 4: Comparisons for enhanced (E) and BERT model (SOTA), underlined text is the key part

Case	Byte (E)	Char (E)	Subword (E)	SOTA	Text
1	0.8174	0.8004	0.2213	0.3329	And Popeye for VP ... F T W.
2	0.8949	0.9574	0.1003	0.1775	I've played a lot of football,and my brother played quarterback in the NFL.So suck it,W a n k e r.
3	0.8017	0.0156	0.0061	0.3189	HAHAHAHA...chucky, you're toast, 💩 fer brains.
4	0.9283	0.1561	0.0028	0.1676	I throw soo much stuff out there. I am one big walking gimmick and guess what, you bought into me a lonnnnng time ago ⓢⓤⓒⓚⓔⓡ.
5	0.0960	0.3049	0.8882	0.0832	Obama hates me so I hates him back. But unlike him, I love America. He gets my middle finger, on both hands.
6	0.1168	0.1833	0.6912	0.3689	India is a toilet...a smelly one at that.
7	0.8041	0.9399	0.8398	0.1003	Isn't this the same FUCKING BITCH that said "at this point, what does it matter" with BENGHAZI?!!
8	0.4063	0.0770	0.7853	0.1881	Phuque all Abrahamic BASED religions!

5.4.4 Case studies

In this section, we dive deep into different textual granularity for enhanced BERT models through some case studies as reported in Table 4.

Table 5: Comparisons between MGEL and SOTA

Source	Method	AUC@ROC	AUC@PR	MCC	F1 Score
Finance	SOTA	94.90	51.33	46.87	47.02
	MGEL	96.02 (↑1.12)	56.96 (+5.63)	52.10 (↑5.23)	53.21 (↑6.19)
News	SOTA	95.53	72.76	65.25	66.85
	MGEL	95.91 (↑0.38)	74.27 (↑1.51)	65.20 (↓0.05)	66.16 (↓0.69)
Wikipedia	SOTA	97.82	89.32	78.37	80.63
	MGEL	98.24 (↑0.42)	90.58 (↑1.26)	79.73 (↑0.36)	81.92 (↑1.29)
Twitter	SOTA	91.57	84.83	67.39	77.26
	MGEL	93.78 (↑2.21)	88.56 (↑3.73)	71.82 (↑4.43)	79.89 (↑2.63)

Cases 1-2 show that the subword models are not good at intentional misspellings in comparison to both byte and character ones. The obfuscation, however, could be easily defused by byte and character models since these characters stand together. Cases 3-4 further demonstrate that the byte model could be more powerful than the character model for Emojis and special multi-byte characters (e.g., three-byte ⓢ). Although they are indeed included in the vocabulary of character and subword models, multi-byte characters (four-byte emojis) are not likely to get trained reasonably for a good embedding due to limited samples involving the same emojis. The byte model, however, is able to learn a good embedding of partial bytes of the whole multi-byte characters. This is related to the character encoding where similar ones are usually standing together and have many common bytes. For instance, different emoji smileys have common 3 head bytes ['0xf0', '0x9f', '0x98'][8]. Cases 5-6 are good examples that BERT model is a context-aware language model. Specifically, all words are not abusive, but the whole sentence or the combination of multiple words is offensive. Lastly, cases 7-8 show that it's necessary to develop enhanced models from scratch for abusive language.

5.4.5 MGEL performance

Byte and character models are able to detect some intentionally manipulated challenging cases, albeit being inferior to subword ones overall. In this context, we resort to MGEL to integrate NBLR and different Enhanced BERT models. The ensemble probability is denoted as $\mathrm{p} = r_3 * \left[r_1 * \mathrm{p(byte)} + r_2 * \mathrm{p(char)} + (1 - r_1 - r_2) * \mathrm{p(subword)}\right] + (1 - r_3) * \mathrm{p(nblr)}$ where weights $r_1, r_2, r_3 \in [0, 1]$. We search the weight space on development set with step size 0.1. The overall weights $r_1 = 0.2, r_2 = 0.2, r_3 = 0.9$ are applicable for all datasets. The performance comparisons are detailed in Table 5. MGEL marks new state-of-the-art performance for abusive language detection overall.

[8]https://getemoji.com/

6 Discussion and Outlook

Although NBLR is inferior to BERT models, the throughput (request per second) is multiple times higher due to the simplicity when they were deployed in a service. For one-hot character encoding on CNN and BiLSTM, we also experiment with increasing vocabulary sizes (e.g, 200, 300) to include more useful multi-byte characters based on odds ratios but it doesn't work well. This is also our initial motivation for byte-level explorations.

The enhanced changes that are applied to BERT are simple to implement and work well in practice. We are well ~~definitely~~ aware that a large amount of studies on BERT improvement. However, ~~beating~~ competing with the existing BERT variants for generic language modeling is not our focus here despite they could be potentially applied to our problem. For example, ERNIE aims to infuse knowledge into BERT model by masking predefined entities and phrases implicitly (Sun et al., 2019), which is somewhat similar to our bi-gram whole-word masking. Our work, instead, focuses on abusive language understanding and detection itself. The multi-grained text decomposition analysis also shows that a single language model cannot cover all aspects of abusive languages.

Recently, byte-level subwords have also been used in language modeling (Liu et al., 2019; Wang et al., 2019). In machine translation, byte-level BPE enables multi-lingual representation more compact (Wang et al., 2019) and delivers better performance. In addition to subword, character and byte, we also experiment byte-level subword based on unigram language model (Kudo, 2018) BERT models. Unfortunately, the performance in our datasets is not as appealing as those reported in machine translation (Wang et al., 2019).

7 Conclusion

The multi-grained text analysis indicates that byte and subword outperform character and word respectively in almost all cases. BiLSTM could also boost performance of word and subword inputs but deteriorate byte and character ones compared to CNN. NBLR delivers a competitive performance even against deep learning models. More importantly, we proposed an ensemble model, MGEL, that offers the best performance on the largest abusive language datasets, and significantly improves over the state-of-the-art hatespeech detection algorithms.

References

Sweta Agrawal and Amit Awekar. 2018. Deep learning for detecting cyberbullying across multiple social media platforms. In *European Conference on Information Retrieval*, pages 141–153. Springer.

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.

Sravan Bodapati, Spandana Gella, Kasturi Bhattacharjee, and Yaser Al-Onaizan. 2019. Neural word decomposition models for abusive language detection. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 135–145.

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

Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. 2012. Detecting offensive language in social media to protect adolescent online safety. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Conference on Social Computing (SocialCom)*, pages 71–80. IEEE.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Jesse Davis and Mark Goadrich. 2006. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Antigoni Maria Founta, Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Athena Vakali, and Ilias Leontiadis. 2019. A unified deep learning architecture for abuse detection. In *Proceedings of the 10th ACM Conference on Web Science*, pages 105–114. ACM.

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.

Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2015. Multilingual language processing from bytes. *arXiv preprint arXiv:1512.00103*.

Tommi Gröndahl, Luca Pajola, Mika Juuti, Mauro Conti, and N Asokan. 2018. All you need is" love" evading hate speech detection. In *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*, pages 2–12.

Kazuki Irie, Pavel Golik, Ralf Schlüter, and Hermann Ney. 2017. Investigations on byte-level convolutional neural networks for language modeling in low resource speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5740–5744. IEEE.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *AAAI*, pages 2741–2749.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Kai Chen, Gregory S Corrado, and Jeffrey A Dean. 2015. Computing numeric representations of words in a high-dimensional space. US Patent 9,037,464.

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.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Manuel Perea, Jon Andoni Duñabeitia, and Manuel Carreiras. 2008. R34d1ng w0rd5 w1th numb3r5. *Journal of Experimental Psychology: Human Perception and Performance*, 34(1):237.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. *SocialNLP 2017*, page 1.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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)*, pages 1715–1725.

Sara Owsley Sood, Judd Antin, and Elizabeth F Churchill. 2012. Using crowdsourcing to improve profanity detection. In *AAAI Spring Symposium: Wisdom of the Crowd*, volume 12, page 06.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2019. Neural machine translation with byte-level subwords. *arXiv preprint arXiv:1909.03341*.

Sida Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2*, pages 90–94. Association for Computational Linguistics.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

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.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

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