Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 202–210
Marseille, 20-25 June 2022

Identification of Multiword Expressions in Tweets for Hate Speech Detection

Nicolas Zampieri[1], **Carlos Ramisch**[2], **Irina Illina**[1], **Dominique Fohr**[1]
[1]Lorraine University, CNRS, Inria, Loria, F-54000 Nancy, France
[2]Aix-Marseille University, Université de Toulon, CNRS, LIS, Marseille, France
nicolas.zampieri@inria.fr, {illina, fohr}@loria.fr, carlos.ramisch@lis-lab.fr

Abstract

Multiword expression (MWE) identification in tweets is a complex task due to the complex linguistic nature of MWEs combined with the non-standard language use in social networks. MWE features were shown to be helpful for hate speech detection (HSD). In this article, we present joint experiments on these two related tasks on English Twitter data: first we focus on the MWE identification task, and then we observe the influence of MWE-based features on the HSD task. For MWE identification, we compare the performance of two systems: lexicon-based and deep neural networks-based (DNN). We experimentally evaluate seven configurations of a state-of-the-art DNN system based on recurrent networks using pre-trained contextual embeddings from BERT. The DNN-based system outperforms the lexicon-based one thanks to its superior generalisation power, yielding much better recall. For the HSD task, we propose a new DNN architecture for incorporating MWE features. We confirm that MWE features are helpful for the HSD task. Moreover, the proposed DNN architecture beats previous MWE-based HSD systems by 0.4 to 1.1 F-measure points on average on four Twitter HSD corpora.

Keywords: multiword expressions, hate speech, social media

1. Introduction

A multiword expression (MWE) is a lexicalised combination of two or more lexemes which exhibits some form of idiomaticity (Baldwin and Kim, 2010). Automatic identification of MWEs is a difficult task in natural language processing because, among others, MWEs can have discontinuities and overlaps (Constant et al., 2017). Moreover, only a few corpora annotated in terms of MWEs are available.

In this article, we study the robustness of MWE identification systems on non-standard texts, namely tweets. Indeed, tweets often employ non-standard syntax and contain spelling mistakes, abbreviations, etc. We hypothesise that, under these conditions, the MWE identification task becomes even more difficult.

Hate speech is commonly defined as *a communication that disparages a person or a group based on some characteristic such as race, colour, gender, etc.* (Nockeby, 2000). Manual moderation of harmful tweets is not possible due to the huge number of tweets posted every day. Thus, automatic methods to support social media moderation can potentially help fight online harassment, cancellation, polarisation, misinformation, etc.

In this work, we are interested in studying the impact of different MWE identification systems for automatic hate speech detection (HSD). Previously, Stanković et al. (2020) and Zampieri et al. (2021) have shown that MWEs are helpful for this task. We compare two automatic MWE identification systems: the first one utilises a look-up method on a lexicon, the second one is based on a deep neural network (DNN). The identified MWEs are employed as additional features in a newly proposed DNN architecture for HSD.

We structure the article as follows. Related work in the field of MWE identification and HSD is presented in Section 2. Our study on MWE identification on tweets is described in Section 3. Section 4 highlights the impact of MWE features on HSD. Finally, we conclude and propose directions for future work.

2. Related Work

MWE identification is defined as automatically *annotating* MWE occurrences in a corpus (similar to named entity recognition). MWE identification should be distinguished from MWE discovery, which consists in *extracting* a list of MWEs from corpus (Constant et al., 2017). MWE discovery is not covered in this paper.

The MWE identification task has been addressed in the past with statistical sequence tagging models, e.g., conditional random fields – CRFs (Constant et al., 2012) and structured perceptron (Schneider et al., 2014). Parsing-based models have also been employed, such as tree-substitution grammars (Green et al., 2013) and dependency transition-based parsing (Constant and Nivre, 2016). MWE identification has also been accomplished using dictionaries and rule-based systems such as the mwetookit (Cordeiro et al., 2016).

The systems submitted to recent shared tasks led to advances in the state of the art (Schneider et al., 2016; Savary et al., 2017; Ramisch et al., 2018; Ramisch et al., 2020). The best system in the PARSEME shared task 2017, named Transition, was adapted from Constant and Nivre (2016) using a transition-based parsing system. In 2018, the best system TRAVERSAL was a tree CRF (Waszczuk, 2018), although some neural models performed quite well, e.g., TRAPACC (Stodden et al., 2018). The 2020 edition benefited from advances

in pre-trained language models, as exemplified by the best system, MTLB-struct, based on a BERT model fine-tuned using a multi-task parsing and MWE identification objective (Taslimipoor et al., 2020).

The lexical-semantic recognition system of Liu et al. (2021) is a recent BERT-based system which predicts MWEs and supersense tags using a single supertag system. It consists of a recurrent neural network that takes as input frozen contextual embeddings from BERT. The system obtained impressive results on the Streusle corpus (Schneider and Smith, 2015) and was also evaluated cross-domain on the PARSEME English corpus (Ramisch et al., 2018) and on DimSum (Schneider et al., 2016). We utilise this system in our experiments given that it is recent, simple, well documented and freely available.

Some papers have analysed the performance of MWE identification. Maldonado and QasemiZadeh (2018) showed that MWE identification performance is closely related to the rate of unseen MWEs in the test set. Savary et al. (2019) argue that lexicons are needed to obtain better generalisation of MWE identification, where generalisation is harder than in similar tasks such as named entity recognition. The evaluation of MWE identification in downstream tasks is quite rare, and we discuss it specifically for HSD below.

Hate speech detection is a challenging task in the field of natural language processing. Early approaches were based on features with classifiers such as support vector machines and logistic regression. Waseem and Hovy (2016) employed character-level features with logistic regression to classify tweets. Davidson et al. (2017) classified tweets using word-level features, part-of-speech, sentiment and meta-data of tweets with a logistic regression classifier. Other hard-coded features have been used for hate speech detection, such as user features (Fehn Unsvåg and Gambäck, 2018). A survey that summarises the state-of-the-art features has been done by Schmidt and Wiegand (2017).

Recently, most HSD systems are based on DNNs with word embeddings. Badjatiya et al. (2017) showed that DNN approaches outperform state-of-the-art character/word n-gram approaches. Gambäck and Sikdar (2017) proposed a convolutional neural network system that outperforms a logistic regression classifier. Zhang et al. (2018) proposed a DNN architecture based on convolutional and recurrent neural networks. Cao and Lee (2020) proposed the HateGAN system, which uses an adversarial method based on reinforcement learning and shows important improvements on HSD. Awal et al. (2021) developed the AngryBERT system, which was trained for hate speech detection and sentiment classification.

Multiword expressions and hate speech detection have been the focus of a couple of recent studies. Stanković et al. (2020) extended a Serbian lexicon of abusive language with special attention to MWEs and proposed to exploit it to create an abusive-language corpus for the Serbian language. Zampieri et al. (2021) developed a DNN-based system that uses MWE features. The MWE features were integrated in a DNN-based system that utilises the categories of MWEs. These two works have shown that MWEs are helpful for HSD.

3. MWE Identification in Tweets

In this section, we explain our methodology for MWE identification in tweets, and present its experimental evaluation results.

3.1. Methodology

The goal of the automatic MWE identification task is to tag the words that belong to MWEs. We analyze the robustness of two MWE identification systems for tweets: a **lexicon-based** approach based on the *mwetoolkit* (Cordeiro et al., 2016), and a **lexical recognition system** (LSR) based on a DNN (Liu et al., 2021).

For the lexicon-based approach, we extract a list of MWEs from several annotated corpora. Each word of the extracted MWE is lemmatised and the canonical forms of extracted MWEs are put in the lexicon of MWEs. The lexicon contains both MWEs that appear in contiguous configurations (e.g., *I returned to* ***pick up*** *my car*) and non-contiguous configurations (e.g., *I* ***picked*** *it* ***up*** *when it was finished*) in the annotated corpora. For the latter, only the words composing the MWE are kept, ignoring intervening words (e.g., both instances above will yield a single entry ***pick up*** in the lexicon).[1] The lexicon is then projected on the test corpus to annotate the MWEs, as detailed in Section 3.2.

The LSR system is based on DNNs and should have a higher capability of generalisation from the examples compared to the lexicon-based system. The LSR architecture consists of a BERT model (Devlin et al., 2019), followed by two bidirectional long short-term memory (Bi-LSTM) layers and one CRF layer. We use this system in our experiments given that it is recent and obtained good results in cross-domain evaluations.

We are interested in studying different training configurations of the LSR system: varying the amount and the nature of the training set and using different "BIO" tagging schemes (see Figure 1). The "BIObio" scheme is similar to the original BIO tagging scheme with MWE categories and supersenses proposed by Liu et al. (2021). Each token is tagged "B" if it is at the beginning of a MWE, "I" if it is inside a MWE, "O" if it does not belong to a MWE. The labels "b","i" and "o" have the same meaning as "B", "I" and "O" labels, but the tagged MWE is nested within an encompassing MWE. In "BIObio", lexical and MWE categories (e.g., VID for verbal idioms, VPC for verb-particle constructions) are concatenated with the initial tags "B" and "b". Tokens different from "I" and "i" are also concatenated to lexical categories (e.g., noun, verb) and, if applicable,

[1] Lexicon entries are not reordered, e.g., ***take pictures*** and ***pictures taken*** are extracted as two distinct lexicon entries.

to supersenses. The "BIOo-cat" tagging scheme concatenates the lexical and MWE categories to the labels "B" and "I", but not to "O" labels. The "BIOo" tagging scheme is even simpler and has no MWE categories. Differently from Liu et al. (2021), these two schemes (BIO-cat and BIOo) ignore supersenses.
The LSR system can predict a structurally invalid tagging: e.g., a word tagged with the label "I" can appear before a word tagged with a label "B" in a sentence. To correct the invalid sequences of predictions of "BIO" labels, we apply a filtering on the outputs of the LSR system as detailed below.

3.2. Experimental Setup

In this section, we describe the corpora for the MWE identification task and the configurations of our systems.

Corpora	Sets	#sent.	#tokens	#MWEs
Streusle	Train	2,724	44,822	2,425
	Dev	554	5,394	283
	Test	535	5,381	281
PARSEME	Train	3,471	53,201	331
	Test	3,965	71,002	501
Tweet part of DimSum	Train	987	18,247	1,112
	Test	500	6,627	362

Table 1: Number of sentences, tokens and strong MWE occurrences in the standard partitions in training, development, and test sets for Streusle, PARSEME and DimSum corpora.

Table 1 shows the statistics of three English corpora.
Streusle is a corpus of online reviews (non-tweets) annotated in terms of weak (e.g., *narrow escape*, *do not be surprised*) and strong (e.g., *go out of my way*, *close call*) MWEs and supersenses (Schneider and Smith, 2015). MWES in the corpus are annotated into 20 fine-grained categories and divided into training, validation and testing sets. We employ version 4.3 of the Streusle corpus.
The **PARSEME** corpus (Ramisch et al., 2018) does not contain tweets and is annotated only in terms of strong verbal MWEs. Six categories of verbal MWEs are considered. The English PARSEME corpus is only available in version 1.1 and is split in training and test sets, with no development set.
The **DimSum** corpus (Schneider et al., 2016) contains online web reviews, TED talk transcripts, and tweets. In our work we use only the tweet part of this corpus because we focus our experiments on tweets, as the corpora used in HSD experiments contain only tweets (Section 4). The corpus is annotated in terms of strong MWEs using binary labels: a word either belongs to a MWE or not. We exploit the test part of this corpus as **test set** for assessing our MWE identification systems. All MWE identification system configurations are evaluated on the tweets contained in the test part of DimSum.
As PARSEME and DimSum corpora are annotated in terms of strong MWEs, we take into account only strong MWE annotations from the Streusle corpus. Weak MWEs are not taken into account (except for the LSR_1 configuration).
For all corpora except the DimSum test set, we "normalise" MWEs: in a given sentence, when a word is common to two MWEs (MWE overlap) or if two MWEs are nested, we remove the second MWE.[2] This phenomenon is infrequent and occurs in less than 5% of sentences, so this normalisation can be performed without significantly impoverishing the training data.
For the lexicon-based configuration, we extract MWEs from all the above corpora, except the DimSum test set: Streusle train, dev and test, PARSEME train and test, DimSum train. The obtained lexicon contains 3,255 MWEs. We utilise the DimSum training set to tune the parameters of the lexicon-based system. We evaluated the use of parts-of-speech with the lemmas of MWE component words. Parts of speech do not show improvement in MWE identification on the development set. Thus, we use lemmas only. We also experimented several values to tune the maximal gap length between words composing MWEs when they are discontinuous. The optimal value of 3 is chosen in the following experiments.
For the LSR model, we train seven configurations. We recall that the proposed LSR configurations differ in the training data and the granularity of tagging labels. We train each configuration five times with different random seeds for initialisation. We use early stopping with 10 epochs for patience.
LSR_1 configuration corresponds to the system proposed in Liu et al. (2021). In this configuration, we train the LSR model on the Streusle training set and utilise the default labelling scheme as in Liu et al. (2021), with weak and strong MWE labels. It is a complex tagging scheme, and counts around 600 labels.
LSR_2 configuration is also trained on the Streusle training set. We adopt the"BIOo-cat" tagging scheme. Compared to the LSR_1 configuration, weak MWEs are ignored, the supersense labels are omitted, as well as lexical categories in non-MWE tags. The final number of labels is 42.
LSR_3 configuration is also trained on the Streusle training set. We utilise the "BIOo" tagging scheme and predict only 4 labels. The goal of LSR_1, LSR_2 and LSR_3 configurations is to study the impact of different labelling schemes on MWE identification.
LSR_4 configuration is trained on the DimSum training set. As the DimSum corpus has no fine-grained categories, we use the "BIOo" labelling scheme with 4 labels. This system uses only the limited in-domain data available.
LSR_5 configuration is trained on the concatenation of the DimSum (tweets) and the Streusle (non-tweets) training sets. We utilise the "BIOo" tagging scheme with

[2] We remove the MWE whose first token appears later, or the shortest one if they start at the same position.

BIOo	O	B	o	o	I	O	O	B	I	O
BIOo-cat	O	B-V.LVC.full	o	o	I-V.LVC.full	O	O	B-N	I-N	O
BIObio	O-PRON	B-V.LVC.full.v.social	o-DET	o-ADJ	I_	O-P-p.purpose	O-DET	B-N-n.body	I_	O-PUNCT
Sentence	I	**had**	a	routine	**surgery**	for	an	**ingrown**	**toenail**	.

Figure 1: Example of BIO labelling for the LSR model. This example has two MWEs: ***had surgery*** and ***ingrown toenail***. The first token of each MWE is tagged 'B' (begin), the following MWE tokens are tagged 'I' (inside). For BIOo-cat and BIObio, the categories are appended to the tags: lexical category (e.g., 'V' for verbal, 'N' for nominal), and MWE category (e.g., 'LVC-full' for full light-verb constructions). In BIObio, supersenses (e.g., 'n.body' for body parts) are added to the tags, and lexical categories are also appended to 'O' (outside) tags but not to 'I' tags, as in Liu et al. (2021).

4 labels, as in LSR_4. The goal of this system is to verify whether completing the in-domain data of DimSum with out-of-domain data from Streusle helps.

LSR_6 configuration is the union of predictions from two sub-systems. The first one is trained on the PARSEME and Streusle training sets and covers only verbal MWEs and 14 labels. The second one is trained on the Streusle training set to predict non-verbal MWEs (30 labels). This configuration uses the "BIOo-cat" tagging scheme for both sub-systems. If the final prediction, resulting from the union of the predictions of both sub-systems, has an MWE overlap, we choose to keep the MWE whose first token appears first. The idea here is to make use of the maximum of out-of-domain data available: Streusle for all MWEs, plus the extra annotations for verbal MWEs from PARSEME.

LSR_7 configuration is the same configuration as LSR_6 except for the label set. In this configuration, we adopt "BIOo" tagging scheme and 4 labels.

Other configurations are not possible to train because some corpora do not have category and supersense annotations. For each of the LSR configurations described above, we employ the **Streusle development set** to tune the filtering parameters. We evaluated different heuristics to filter the LSR outputs and adopted the following ones: we remove single-token MWEs, "I" labels not preceded by a "B" label, and MWEs containing special tokens (@USER, URL, and hashtags). The MWE maximum gap length was also tuned and set to 2, removing all MWEs containing gaps greater than 2.

Evaluation metrics. To evaluate the MWE identification systems, we adopt standard metrics which were applied for the PARSEME (Savary et al., 2017) and DimSum (Schneider et al., 2016) shared tasks. The *MWE-based measure* is the F1-score for fully predicted MWEs. The *token-based measure* is the F1-score for tokens belonging to a MWE, assessing partial matches. The *MWE-link-based measure* is the F1-score based on matching adjacent word pairs within MWEs, and gives credit to partly correct MWEs without accounting for single-token predictions.

3.3. Results

In this part, we present results obtained for the MWE identification task on tweets part of the DimSum test set. Table 2 shows that the lexicon-based system achieves 28.7% MWE-based F1-score. This performance can be due to the fact that 78% of the MWEs present in the DimSum tweets test set are not present in the created lexicon of MWEs. The lexicon-based approach cannot find MWEs not present in the lexicon of the system. In other words, although reasonably precise, the lexicon-based system is unable to generalise and obtains poor recall, especially given that most of the corpora from which it was extracted is out of domain.

All LSR configurations outperform the lexicon-based approach. We observe that LSR_{2-7} configurations improve both recall and precision (in terms of MWE-based F1 measure) compared to the lexicon-based approach. This suggests that LSR configurations generalise and detect MWEs that are not present in the same form in the training set.

LSR_5 achieves the best results in terms of MWE-based, token-based and MWE-link-based F1-scores. Comparing token-based F1-scores of LSR_5 and of the lexicon-based system (56.8% versus 28.5%), we observe that the LSR_5 system predicts partial MWEs better.

In order to study the impact of the tagging schemes, we compare three LSR configurations trained on the same corpus with different tagging schemes: LSR_1, LSR_2 and LSR_3. They are trained on Streusle training set with "BIObio", "BIOo-cat" and "BIOo" tagging schemes. From Table 2, we observe that the complex "BIObio" tagging obtains lower F1 scores. Indeed, the LSR_1 system using "BIObio" obtains 36.1% MWE-based F1 score compared to 43.3% achieved by LSR_2 or LSR_3 systems. We observe the same performance for configurations using "BIOo-cat" and "BIOo" tagging, which indicates that adding the MWE categories does not help the system. This is confirmed by the results obtained by LSR_6 and LSR_7.

Now, we focus our observation on the configurations using the same tagging schemes and different training sets: LSR_3, LSR_4, LSR_5 and LSR_7. We observe that LSR_4 has the lowest F1 scores, reaching 41.2% MWE-based score. This can be due to the fact that it utilises

Configurations (train corpus)	Labels	MWE-based Precision	MWE-based Recall	MWE-based F1-score	Token-based F1-score	MWE-link-based F1-score
Lexicon-based	-	45.5	21.0	28.7	28.5	25.9
LSR_1 (ST)	BIObio	45.5 ± 3.4	29.9 ± 2.0	36.1 ± 2.4	47.6 ± 1.3	43.8 ± 1.4
LSR_2 (ST)	BIOo-cat	53.7 ±1.1	36.4 ±2.6	43.3 ±1.6	53.5 ±2.1	51.2 ±2.1
LSR_3 (ST)	BIOo	49.0 ±2.7	39.2 ±4.1	43.3 ±1.5	54.7 ±1.8	52.0 ±2.0
LSR_4 (DSM)	BIOo	**61.1** ±2.7	31.2 ±2.6	41.2 ±2.3	51.8 ±3.1	48.5 ±2.7
LSR_5 (ST-DSM)	BIOo	60.4 ±2.5	37.9 ±0.9	**46.5** ±0.3	**56.8** ±1.0	**54.0** ±1.5
LSR_6 (ST-PSM)	BIOo-cat	53.2 ±1.5	37.1 ±2.0	43.6 ±1.3	54.1 ±1.7	50.9 ±1.7
LSR_7 (ST-PSM)	BIOo	50.0 ±4.4	**39.9** ±3.3	44.1 ±0.9	54.7 ±2.1	51.9 ±2.4

Table 2: MWE identification results on the DimSum test tweet set. For each result, the average score and the standard deviation of 5 runs are given (except for the lexicon-based configuration). "ST", "DSM" and "PSM" stand for Streusle, DimSum and PARSEME, respectively. "Labels" represents the BIO labelling scheme for LSR.

the smallest training set which contains 987 sentences, compared to LSR_3, LSR_5 and LSR_7 systems which are trained on more than 2,724 sentences. The LSR_7 system, which is trained on the Streusle and PARSEME corpora, does not improve the F1 score compared to the LSR_3 system, which uses only the Streusle training set. This can be due to the fact that the LSR_7 system utilises two DNN models trained independently. The LSR_5 system, which is trained on the concatenated Streusle and DimSum training sets, achieves the best F1 scores: 46.5%, 56.8% and 54.0% of MWE-based, token-based and MWE-link-based F1 scores, respectively. This indicates that a single system trained on both in- and out-of-domain data can probably benefit from both sources of information, as the LSR_5 system is trained on tweets and non-tweets data.

Our experiments suggest that training an LSR system on tweets and non-tweets data with the "BIOo" tagging scheme is the best configuration for the MWE identification task on tweets and it outperforms the lexicon-based approach. This is an encouraging result for the following experiments, as we will see in the next section.

4. Hate Speech Detection with MWE Features

Zampieri et al. (2021) show that MWE features, provided by a lexicon-based MWE identification system, improve HSD results. In this section, we study the impact of LSR MWE identification systems for the HSD task and compare it with the lexicon-based MWE system described previously.

4.1. Methodology

To study the impact of MWE features for the HSD task, we utilise two of the MWE identification systems presented in Section 3: the lexicon-based system and the best LSR configuration (LSR_5).

To integrate MWE features in the hate speech detection system, we study two architectures of HSD. The first HSD architecture, named *HSD-3B*, was proposed by Zampieri et al. (2021) and is composed of three branches of neural networks. One branch takes into account an entire sentence, embedded with the Universal Sentence Encoder – USE (Cer et al., 2018). Two other branches deal with MWE features: one branch embeds the MWE category of each word in the sentence and is followed by convolutional layers; and the other branch contains the word embedding of each word composing the MWEs of the sentence (words that do not belong to a MWE are not used) and is followed by a bidirectional LSTM (Bi-LSTM) layer. This latter branch allows to better represent the contents of the MWEs. The outputs of the three branches are concatenated and are followed by two dense layers.

The second HSD architecture, named *HSD-2B*, is proposed in this work and consists of two branches as presented in Figure 2. The first branch is dedicated to the USE sentence embedding as in the HSD-3B system. The second branch uses word embeddings of all words of the sentence, concatenated with their corresponding MWE categories. In this architecture we give more information to the system (embeddings of all words) compared to the HSD-3B system. To take into account past and future context information of each word, a Bi-LSTM layer is added. The outputs of the two branches are concatenated and are followed by two dense layers as in the HSD-3B system.

We compare the HSD-2B and HSD-3B systems with a *baseline* system. The baseline system employs only sentence embeddings (USE) as input and is made up of two dense layers, without MWE features.

MWE features. The lexicon-based and the LSR_5 MWE systems predict, for each word of a tweet, whether it is part of a MWE or not. The fine-grained categories of MWEs are not available for these two MWE identification systems. Thus, we see the prediction of MWEs as binary MWE categories. For the LSR_5 MWE predictions, we transform the "BIOo" labels into binary labels as follows: "B" and "I" labels are transformed into ones, "O" and "o" labels transformed into zeroes.

4.2. Experimental Setup

We use four Twitter hate speech corpora for evaluation. The **Waseem** corpus (Waseem and Hovy, 2016) contains 16,919 tweets annotated in three classes: sexist, racist and neither. We focus on the HSD task, so we group

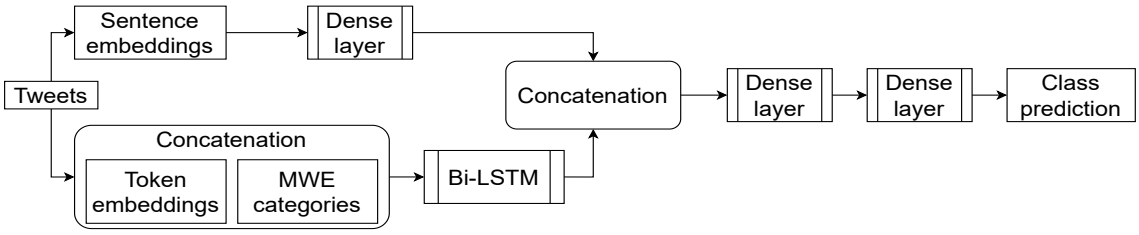

Figure 2: Proposed HSD-2B system with two branches.

together sexist and racist classes into one class (hateful). Tweets labelled as "neither" are labelled as non-hateful. The corpus contains 73% of non-hateful tweets and 27% of hateful tweets.

The **Davidson** corpus (Davidson et al., 2017) is a tweet corpus annotated in terms of hate speech, offensive speech or neither. The corpus contains 24,802 tweets: 76% are offensive, 11.4% are hateful, and 16.6% are neither.

The **Founta** corpus (Founta et al., 2018) contains 100k tweets annotated in four classes: hate speech, abusive speech, normal speech and spam. Our experiments focus on HSD, so we remove spam tweets and keep around 86k tweets. The corpus contains 63% of normal tweets, 31% of abusive tweets and 6% of hateful tweets.

The **HatEval** corpus (Basile et al., 2019) is provided by the SemEval2019 shared task 5. It contains 13k tweets annotated as hateful and non-hateful speech. It is a balanced corpus with 42% hateful and 58% non-hateful tweets.

For the Waseem, Davidson, and Founta datasets, we utilise 60%, 20% and 20% as **training, validation, and test sets**, respectively. For HatEval corpus, we use the standard corpus partition into training, development and test sets with 9k, 1k and 3k tweets, respectively. For Waseem and HatEval, the HSD task is a binary classification, whereas for the other corpora, it is ternary.

We apply the following pre-processing for each tweet of all corpora: we remove mentions, hashtags, URLs and we replace emojis with readable text (e.g., ♥ → *:heart:*). To tag MWEs with the lexicon-based system, we lemmatise tweets with the *spacy-udpipe* python library.

Hyperparameters of HSD systems. All systems (baseline, HSD-2B and HSD-3B) utilise USE embeddings of size 512. As word embeddings, we use the BERTweet contextual token embeddings (Nguyen et al., 2020) of size 768. BERTweet uses a tokeniser that splits some words in several sub-words (e.g, *playing* → *play @ing*). We set the maximal length of tweets to 128 tokens. For the HSD-3B system, we use the same hyper-parameters as in Zampieri et al. (2021). For the HSD-2B system, we set the dimensions of the Bi-LSTM and all dense layers to 128 and 256 neurons, respectively.

Evaluation metrics. We evaluate our models in terms of macro-average F1. It is the average of the F1 scores across all classes.

4.3. Results

The goal of our experiments is to study the impact of MWE features on a HSD system, and to compare the lexicon-based and the LSR_5 MWE identification systems for the HSD task. First, we analyse MWE identification in the target hate speech corpora. Second, we compare the system with and without MWE features. Finally, we compare the lexicon-based system with the LSR_5 MWE identification system for the HSD task.

MWE systems	Waseem	Davidson	Founta	HatEval
lexicon	4,578	6,745	31,391	6,040
LSR_5	4,966	10,447	46,679	9,075

Table 3: Number of MWE occurrences tagged by the lexicon-based and the LSR_5 systems in the hate speech training sets.

MWE identification on hate speech corpora. It is important to note that, as HSD corpora are not annotated in terms of MWEs, we have no gold annotations for the MWE identification task. We can only compare the number of MWE occurrences tagged by the two MWE identification systems.

Table 3 displays the number of MWE occurrences tagged by the lexicon-based and by the LSR_5 systems. We observe that the LSR_5 system has tagged more MWEs than the lexicon-based system. We observed similar results on the DimSum tweet test set (see Section 3.2).

HSD systems. We compare three models: the baseline model (without MWE features), HSD-3B and HSD-2B (with MWE features).

Table 4 displays the average macro-F1 of 5 runs on the Waseem, Davidson, Founta, and HatEval test sets. The last column represents the average of the macro-F1 across the four corpora. The baseline system achieves 72.0% of average macro-F1 score. The systems using MWE features outperform the baseline system. This confirms that MWE features are helpful for hate speech detection. Moreover, the HSD-2B system achieves better results on every test corpus compared to the baseline system, with 73.5% of average macro-F1 score.

The HSD-2B system outperforms the HSD-3B system (73.5% versus 72.4%), especially on the Davidson and Founta test sets. This better performance can be due to the fact that HSD-2B has access to the embeddings of all

HSD Systems	MWE ident.sys.	Waseem	Davidson	Founta	HatEval	Average
Baseline	-	79.5 (±0.1)	72.1 (±0.5)	72.4 (±0.7)	64.1 (±0.4)	72.0
HSD-3B	lexicon-based	81.3 (±0.2)	72.2 (±0.9)	71.6 (±0.4)	**66.5** (±0.2)	72.9
	LSR_5	80.8(±0.4)	71.0 (±2.5)	71.8 (±0.6)	66.1 (±0.7)	72.4
HSD-2B	lexicon-based	**82.3** (±0.7)	72.5 (±2.1)	74.0 (±0.6)	65.1 (±0.5)	73.3
	LSR_5	81.9 (±0.6)	**73.3** (±1.4)	**74.1** (±0.7)	64.9 (±1.1)	**73.5**

Table 4: Average macro-F1 and standard deviation of 5 runs of hate speech detection. The *Average* column represents the average of macro-F1 across the four corpora. Underlined results indicate significant improvements compared to the Baseline (Gillick and Cox, 1989). Systems that obtained the median macro-F1 score are used to compute significance.

words of the tweet. In the following, we will continue the analysis only for the best architecture, HSD-2B.

Lexicon-based versus LSR-based MWE identification systems for HSD. To perform a deeper analysis, we compare the influence of the lexicon-based and of the LSR_5 systems on the HSD results for the HSD-2B system. We observe that these two MWE identification systems achieve a similar performances in terms of macro-F1 (73.5% and 73.3%). The lexicon-based system outperforms the LSR_5 system for the Waseem and HatEval corpora and vice versa for the other two corpora. An advantage of the LSR_5 MWE identification system is that larger MWE-annotated corpora will enable a better LSR_5 system, and potentially increase the performance of the HSD task.

Our experiments show that the MWE features are useful for the detection of hate speech. Our experimental evaluation shows that there is no significant difference between the use of a lexicon-based system and the LSR identification system for the HSD task.

5. Conclusions and Future Work

In this work, we studied the performance of lexicon-based and DNN-based MWE identification systems, and the impact of MWE features on the HSD task, focusing on tweet corpora.

We proposed and performed an intrinsic evaluation of 7 configurations for the LSR system. We found that LSR systems outperform the lexicon-based system for the MWE identification task on the DimSum tweets test corpus. The best configuration of LSR system is LSR_5, which is trained on tweets and non-tweets data and uses the most coarse label set.

For the HSD task, we studied the impact of the MWE features using lexicon-based and DNN-based MWE identification systems. We proposed an HSD system with 2 branches of DNNs: the first one uses sentence embeddings and the second one exploits the token embeddings concatenated with the MWE categories for each word.

We performed our experiments on four hate speech tweet corpora. The HSD system with MWE features outperforms the baseline system (without MWE features). Our proposed HSD system with two branches gives better results compared to our previous HSD system with three branches. The performance of the lexicon-based and the DNN-based MWE identification systems for the HSD tasks are similar.

In future work, we would like to combine DNN-based and lexicon-based approaches to increase the generalisation of MWE identification.

6. Acknowledgements

Experiments presented in this article were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organisations. This work has been funded by the French Agence Nationale pour la Recherche, through the SELEXINI project (ANR-21-CE23-0033-01).

7. Bibliographical References

Awal, M. R., Cao, R., Lee, R. K.-W., and Mitrovic, S. (2021). Angrybert: Joint learning target and emotion for hate speech detection. In *PAKDD*.

Badjatiya, P., Gupta, S., Gupta, M., and Varma, V. (2017). Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 759–760, Republic and Canton of Geneva, CHE.

Baldwin, T. and Kim, S. (2010). Multiword expressions. In *Handbook of Natural Language Processing*, pages 267–292. CRC Press, Taylor and Francis Group, Boca Raton, 2nd edition.

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rangel Pardo, F. M., Rosso, P., and Sanguinetti, M. (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, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Cao, R. and Lee, R. K.-W. (2020). HateGAN: Adversarial generative-based data augmentation for hate speech detection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6327–6338, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., St. John, R., Constant, N., Guajardo-Cespedes, M.,

Yuan, S., Tar, C., Strope, B., and Kurzweil, R. (2018). Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium, November. Association for Computational Linguistics.

Constant, M. and Nivre, J. (2016). A transition-based system for joint lexical and syntactic analysis. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 161–171, Berlin, Germany, August. Association for Computational Linguistics.

Constant, M., Sigogne, A., and Watrin, P. (2012). Discriminative strategies to integrate multiword expression recognition and parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 204–212, Jeju Island, Korea, July. Association for Computational Linguistics.

Constant, M., Eryiğit, G., Monti, J., van der Plas, L., Ramisch, C., Rosner, M., and Todirascu, A. (2017). Survey: Multiword expression processing: A Survey. *Computational Linguistics*, 43(4):837–892, December.

Cordeiro, S., Ramisch, C., and Villavicencio, A. (2016). UFRGS&LIF at SemEval-2016 task 10: Rule-based MWE identification and predominant-supersense tagging. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 910–917, San Diego, California, June. Association for Computational Linguistics.

Davidson, T., Warmsley, D., Macy, M., and Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515.

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

Fehn Unsvåg, E. and Gambäck, B. (2018). The effects of user features on Twitter hate speech detection. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 75–85, Brussels, Belgium, October. Association for Computational Linguistics.

Founta, A., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., Vakali, A., Sirivianos, M., and Kourtellis, N. (2018). Large scale crowdsourcing and characterization of twitter abusive behavior. *Proceedings of the International AAAI Conference on Web and Social Media*, 12.

Gambäck, B. and Sikdar, U. K. (2017). Using convolutional neural networks to classify hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90, Vancouver, BC, Canada, August. Association for Computational Linguistics.

Gillick, L. and Cox, S. J. (1989). Some statistical issues in the comparison of speech recognition algorithms. In *International Conference on Acoustics, Speech, and Signal Processing,*, pages 532–535 vol.1.

Green, S., de Marneffe, M.-C., and Manning, C. D. (2013). Parsing models for identifying multiword expressions. *Computational Linguistics*, 39(1):195–227.

Liu, N. F., Hershcovich, D., Kranzlein, M., and Schneider, N. (2021). Lexical semantic recognition. In *Proceedings of the 17th Workshop on Multiword Expressions (MWE 2021)*, pages 49–56, Online, August. Association for Computational Linguistics.

Maldonado, A. and QasemiZadeh, B. (2018). Analysis and insights from the parseme shared task dataset. In *Multiword expressions at length and in depth*, pages 149–176, Berlin, October. Language Science Press.

Nguyen, D. Q., Vu, T., and Tuan Nguyen, A. (2020). BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online, October. Association for Computational Linguistics.

Nockeby, J. T. (2000). Hate speech. In Leonard W. Levy, et al., editors, *Encyclopedia of the American Constitution*, pages 1277–1279. Macmillan 2nd edition.

Ramisch, C., Cordeiro, S. R., Savary, A., Vincze, V., Barbu Mititelu, V., Bhatia, A., Buljan, M., Candito, M., Gantar, P., Giouli, V., Güngör, T., Hawwari, A., Iñurrieta, U., Kovalevskaitė, J., Krek, S., Lichte, T., Liebeskind, C., Monti, J., Parra Escartín, C., QasemiZadeh, B., Ramisch, R., Schneider, N., Stoyanova, I., Vaidya, A., and Walsh, A. (2018). Edition 1.1 of the PARSEME shared task on automatic identification of verbal multiword expressions. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 222–240, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Ramisch, C., Savary, A., Guillaume, B., Waszczuk, J., Candito, M., Vaidya, A., Barbu Mititelu, V., Bhatia, A., Iñurrieta, U., Giouli, V., Güngör, T., Jiang, M., Lichte, T., Liebeskind, C., Monti, J., Ramisch, R., Stymne, S., Walsh, A., and Xu, H. (2020). Edition 1.2 of the PARSEME shared task on semi-supervised identification of verbal multiword expressions. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 107–118, online, December. Association for Computational Linguistics.

Savary, A., Ramisch, C., Cordeiro, S., Sangati, F., Vincze, V., QasemiZadeh, B., Candito, M., Cap, F., Giouli, V., Stoyanova, I., and Doucet, A. (2017). The PARSEME shared task on automatic identification

of verbal multiword expressions. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 31–47, Valencia, Spain, April. Association for Computational Linguistics.

Savary, A., Cordeiro, S., and Ramisch, C. (2019). Without lexicons, multiword expression identification will never fly: A position statement. In *Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019)*, pages 79–91, Florence, Italy, August. Association for Computational Linguistics.

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

Schneider, N. and Smith, N. A. (2015). A corpus and model integrating multiword expressions and supersenses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1537–1547, Denver, Colorado, May–June. Association for Computational Linguistics.

Schneider, N., Danchik, E., Dyer, C., and Smith, N. A. (2014). Discriminative lexical semantic segmentation with gaps: Running the MWE gamut. *Transactions of the Association for Computational Linguistics*, 2:193–206.

Schneider, N., Hovy, D., Johannsen, A., and Carpuat, M. (2016). SemEval-2016 task 10: Detecting minimal semantic units and their meanings (DiMSUM). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 546–559, San Diego, California, June. Association for Computational Linguistics.

Stanković, R., Mitrović, J., Jokić, D., and Krstev, C. (2020). Multi-word expressions for abusive speech detection in Serbian. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 74–84, online, December. Association for Computational Linguistics.

Stodden, R., QasemiZadeh, B., and Kallmeyer, L. (2018). TRAPACC and TRAPACCS at PARSEME shared task 2018: Neural transition tagging of verbal multiword expressions. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 268–274, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Taslimipoor, S., Bahaadini, S., and Kochmar, E. (2020). MTLB-STRUCT @parseme 2020: Capturing unseen multiword expressions using multi-task learning and pre-trained masked language models. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 142–148, online, December. Association for Computational Linguistics.

Waseem, Z. and Hovy, D. (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, San Diego, California, June. Association for Computational Linguistics.

Waszczuk, J. (2018). TRAVERSAL at PARSEME shared task 2018: Identification of verbal multiword expressions using a discriminative tree-structured model. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 275–282, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Zampieri, N., Illina, I., and Fohr, D. (2021). Multiword expression features for automatic hate speech detection. In Elisabeth Métais, et al., editors, *Natural Language Processing and Information Systems*, pages 156–164, Cham. Springer International Publishing.

Zhang, Z., Robinson, D., and Tepper, J. (2018). Detecting hate speech on twitter using a convolution-gru based deep neural network. In Aldo Gangemi, et al., editors, *The Semantic Web*, pages 745–760, Cham. Springer International Publishing.