EMBEDDIA

Cross-Lingual Embeddings for Less-Represented Languages in European News Media

Research and Innovation Action
Call: H2020-ICT-2018-1
Call topic: ICT-29-2018 A multilingual Next generation Internet
Project start: 1 January 2019 Project duration: 36 months

D3.3: Initial cross-lingual comment filtering technology (T3.2)

Executive summary

This deliverable summarises progress to date (M18) on Task T3.2 of the EMBEDDIA project, which aims to develop cross-lingual tools for automatic filtering of user-generated comments in news media. We review existing work in news comment filtering and related tasks in social media, showing that no directly applicable resources exist in the less-resourced EMBEDDIA languages. We summarise our progress in developing new filtering classifiers directly for EMBEDDIA data and languages, and for related tasks for which existing high-quality datasets are available only in well-resourced languages. We describe our work so far in cross-lingual transfer, using resources from WP1 to allow transfer of classifiers between languages, and resulting in good filtering accuracy even with minimal data in less-resourced target languages.

Partner in charge: QMUL

Project co-funded by the European Commission within Horizon 2020 **Dissemination Level**		
PU	Public	PU
PP	Restricted to other programme participants (including the Commission Services)	–
RE	Restricted to a group specified by the Consortium (including the Commission Services)	–
CO	Confidential, only for members of the Consortium (including the Commission Services)	–

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825153

Deliverable Information

Document administrative information	
Project acronym:	**EMBEDDIA**
Project number:	**825153**
Deliverable number:	**D3.3**
Deliverable full title:	**Initial cross-lingual comment filtering technology**
Deliverable short title:	**Initial cross-lingual comment filtering**
Document identifier:	**EMBEDDIA-D33-InitialCrosslingualCommentFiltering-T32-submitted**
Lead partner short name:	**QMUL**
Report version:	**submitted**
Report submission date:	**30/06/2020**
Dissemination level:	**PU**
Nature:	**R = Report**
Lead author(s):	**Matthew Purver (QMUL), Ravi Shekhar (QMUL), Kristiina Vaik (TEXTA)**
Co-author(s):	**Marko Pranjić (TRI), Senja Pollak (JSI), Silver Traat (TEXTA)**
Status:	**__ draft, __ final, _x_ submitted**

The EMBEDDIA Consortium partner responsible for this deliverable has addressed all comments received. Changes to this document are detailed in the change log table below.

Change log

Date	Version number	Author/Editor	Summary of changes made
14/05/2020	v0.1	Matthew Purver (QMUL)	Initial draft.
16/05/2020	v0.2	Marko Pranjić (TRI), Senja Pollak (JSI), Silver Traat (TEXTA)	Updated initial draft.
22/05/2020	v1.0	Matthew Purver (QMUL), Ravi Shekhar (QMUL)	First complete draft.
27/05/2020	v1.1	Kristiina Vaik (TEXTA)	Additions to section 4.2.
05/06/2020	v1.2	Matthew Purver (QMUL)	Draft sent for internal review.
07/06/2020	v1.3	Marko Robnik-Šikonja (UL)	Internal review.
09/06/2020	v1.4	Hannu Toivonen (UH)	Internal review.
09/06/2020	v1.5	Matthew Purver (QMUL)	Revision based on internal reviews.
11/06/2020	final	Nada Lavrač (JSI)	Report quality checked and finalised.
30/06/2020	submitted	Tina Anžič (JSI)	Report submitted.

Table of Contents

List of abbreviations

AUC	Area Under Curve
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Network
ELMo	Embeddings from Language Models
LASER	Language Agnostic SEntence Representations
LSTM	Long Short-term Memory
NER	Named Entity Recognition
NLP	Natural Language Processing
NN	Neural Network
ROC	Receiver-Operator Characteristic
RNN	Recurrent Neural Network
UGC	User-Generated Content
TTK	TEXTA Toolkit

1 Introduction

The EMBEDDIA project aims to improve cross-lingual transfer of language resources and trained models using word embeddings and cross-lingual technologies, with a focus on nine languages: Croatian, English, Estonian, Finnish, Latvian, Lithuanian, Russian, Slovene, and Swedish. Work package WP3 aims to apply EMBEDDIA's cross-lingual advances to help news media companies better serve their audience by understanding and analysing their reactions, and assuring the safety, fairness and integrity of their participation in public internet spaces. In Task T3.2, the focus is on automatic moderation and filtering of user-generated content (UGC), primarily the comments readers post under news articles.

1.1 Objectives and structure of the report

The overall objective of workpackage WP3 is to apply EMBEDDIA's cross-lingual technologies to understand and analyse the reactions of multilingual news audiences. The specific objectives of WP3 are as follows:

- O3.1 Advance cross-lingual context and opinion analysis, via Task T3.1;
- O3.2 Develop cross-lingual comment filtering, via Task T3.2;
- O3.3 Develop techniques for report generation from multilingual comments, via Task T3.3.

The objective of this task T3.2 is therefore to develop cross-lingual methods for comment filtering. Work on user needs in WP6 Task T6.1 has identified automatic comment filtering as a key requirement — helping media partners deal with their need to quickly moderate large volumes of user-generated comments — and identified hate speech/abuse detection and trolling detection as two specific tasks of primary importance within that (see WP6 deliverables D6.3 and D6.5).

Our approach on Task T3.2 has at first been i) to develop classifiers for specific tasks, using the methods from Task T3.1 above, and ii) training on already available and trusted social media datasets; these allow us to establish accuracy levels and methods, but are by their nature monolingual and only exist in well-resourced languages. In this scenario, we showed good performance on hate speech and abuse detection (Pelicon et al., 2019; Miok et al., 2019).

In later stages, we tested the application of these approaches to the more general problem of automated comment moderation, using the new EMBEDDIA comment datasets from partners Styria/Trikoder (Croatian) and Ekspress Meedia (Estonian), labelled with the real decisions made by moderators. Training monolingual classifiers on these resources gives models with reasonable accuracy at replicating human moderator decisions, suitable for use in automating comment filtering/moderation, although the noise associated with such real-world datasets suggests that accuracy could be improved if cleaner, standard datasets could be added to the training (Shekhar et al., 2020). Because such datasets are not available in the EMBEDDIA languages (and in most less-resourced languages), one way to achieve this is via limited manual intervention in a semi-supervised approach; another is to use cross-lingual approaches and rely on datasets from other better-resourced languages. We investigated both approaches.

The semi-supervised approach greatly improves performance, although relies on some expert manual effort. In the cross-lingual approach, we first used classifiers based on existing multilingual BERT and LASER embeddings, testing their ability to transfer when trained on standard English hate speech datasets and tested in other languages, including EMBEDDIA language Croatian; this was successful, but evaluation shows performance is noticeably lower than in the monolingual case (Marinšek, 2019). To improve this, our next step was the use of the new EMBEDDIA cross-lingual BERT models from WP1 T1.2 (Ulčar & Robnik-Šikonja, 2020); these improve cross-lingual transfer significantly, producing classifiers that when trained only on English data give good accuracy on Slovene social media data and on the EMBEDDIA Croatian news comments (Pelicon et al., in preparation). Classifiers based on these models have now been implemented as dockerized components for integration with WP6.

The main contributions presented in this report (in the order of appearance) are as follows:

- Monolingual classifiers for hate speech and abuse detection with good accuracy on on standard datasets in well-resourced languages, including 4th place in the SemEval 2019 OffensEval task (Pelicon et al., 2019).
- Monolingual classifiers for automatic comment filtering, trained from real moderator behaviour and giving reasonable accuracy on EMBEDDIA news media partner comment data in EMBEDDIA project languages (Croatian, Estonian) (Shekhar et al., 2020).
- Cross-lingual classifiers for both standard hate speech datasets and EMBEDDIA news comment data, that can be trained on available standard (e.g. English) datasets and transferred to EMBEDDIA project languages (Slovene, Croatian) (Marinšek, 2019).
- Improved cross-lingual classifiers, using the new cross-lingual BERT models from WP1 T1.2, giving cross-lingual accuracy close to monolingual levels, on EMBEDDIA project languages (Slovene, Croatian) (Pelicon et al., in preparation).
- Implemented multi-lingual classifier code and models, available as dockerized components for integration with the Embeddia Assistant in WP6.

This report is split into 6 further sections. Section 2 summarises the user needs that motivate our work, and related work in filtering news comments and other UGC. In Section 3, we describe our baseline work in offensive language detection using standard, monolingual resources. Section 4 describes our initial, monolingual comment filtering classifiers developed on our news media partner data, and shows how performance can be improved using a semi-supervised approach. Section 5 shows how the classifiers can be made cross-lingual, to give general classifiers which can be transferred between languages. Section 6 then summarises the main concrete outputs of this work (code and papers), and Section 7 summarises our conclusions and main findings, and outlines the plans for further work. The appendices include the papers on which the main content sections are based.

2 Background

This section explains the background to this work, first describing the motivation in terms of the needs of the news media industry for automated comment filtering tools, and then outlining the related state of the art in natural language processing.

2.1 User needs

Work on user needs in WP6 Task T6.1 identified automatic comment filtering as a primary need for news media users, to help media partners deal with their need to quickly moderate large volumes of user-generated comments (see WP6, particularly deliverable D6.5). The primary requirements are summed up by the user stories given in deliverable D6.5, the relevant one repeated here for convenience as Figure 1. This describes the problem that must be solved, and the way in which an ideal future version of the EMBEDDIA tools would be used to do that.

Note that hate speech/abuse detection and trolling detection are two major categories on which filtering can be based, but many other phenomena must also be taken into account. Note also that the ability to label outputs with information about which category a to-be-blocked comment belongs to is important. For the EMBEDDIA partners, the primary languages of interest here are Croatian (all newspapers for Styria/Trikoder) and Estonian (the main language for Ekspress Meedia, although many comments in Russian are also received). (Our Finnish partner STT does not currently handle UGC, but only news article text).

Branko is a moderator at 24sata, the largest-circulation daily newspaper in Croatia. 24sata reaches about 2 million readers daily, and many of them post comments about online articles: on an average day, about 8000 comments come in, spread over several hundred articles. Unfortunately, many comments (usually between 5% and 10%) need to be blocked to prevent them appearing online: they might be offensive, dangerous, or legally compromising. This is Branko's job.

Until now, the task of comment filtering and moderation had to be performed almost entirely manually. This is time-consuming and skilled work: the newspaper has a complex moderation policy, as comments may need blocking for a variety of reasons. Some are irrelevant spam or advertising, some contain disinformation, some are threatening or hateful, some obscene or illegal, some written in foreign languages ... so filtering through them all and making consistent decisions is difficult, especially at peak times when over 1,000 per hour may be coming in. Branko uses a system which flags comments that match a list of blacklisted keywords, but this isn't very accurate and is hard to keep up to date as new topics get discussed. With the current COVID-19 crisis, for example, new kinds of spam, fake stories and ethnically-targeted hate speech emerge very fast, and the word lists can't keep up. That means that Branko largely has to rely on fast reading and experience.

The new EMBEDDIA tools for automated comment moderation have made Branko's job much easier. Comments are filtered in real time, automatically detecting those which are most likely to need blocking, ranking them by severity, and labelling them as to which part of the 24sata policy they seem to break. The final decision is left to Branko, but now he can easily prioritise the worst cases first, and make sure they don't appear on the site, without having to read through all the others. He can then check less severe cases, and can leave unproblematic comments where the classifier is very confident for a less busy time. Branko's final decisions are stored and fed back to the system, so that it learns over time to improve, and to adapt to new vocabulary as new topics and stories develop.

Figure 1: User story from Deliverable D6.5: Comment filtering at 24sata, provided by Croatian EMBEDDIA partner Trikoder (Styria Group).

2.2 Related work on comment filtering

Previous work in news comment filtering for automatic moderation is limited. In the only directly relevant work we are aware of, Pavlopoulos et al. (2017b,a) address the problem using data from a Greek newspaper, Gazzetta. They use a dataset of 1.6M comments with labels derived from the newspaper's human moderators and journalists; they test a range of neural network-based classifiers and achieve encouraging performance with AUC scores (area under the ROC curve) of 0.75-0.85 depending on the data subset. However, their data is not directly usable as a training set for our task. Firstly, it is in a different language, and one for which few supplementary resources are available (Greek). Secondly, their moderation labels are binary, representing a "block or not" decision, rather than giving any further information about the reasons behind a decision (see above)[1]. Thirdly, and more fundamentally, each newspaper has its own moderation policy, and the decisions of Gazzetta's moderators are unlikely to be based on the same aims or policies as the decisions we try to simulate for our media partners.

Other work with reader comments on news (see Table 1) exists but does not attempt to learn from or reproduce moderation decisions directly in the same way. Kolhatkar et al. (2019) and Napoles et al. (2017) investigate constructivity in comments, and provide datasets which distinguish between constructive and non-constructive comments; these datasets are related to our task, as they also include information about toxicity and related categories such as insults and off-topic posting. Barker et al.

[1] Pavlopoulos et al. (2017a) asked additional annotators to classify comments according to a more detailed taxonomy (*"We also asked the annotators to classify each snippet into one of the following categories: calumniation (e.g., false accusations), discrimination (e.g., racism), disrespect (e.g., looking down at a profession), hooliganism (e.g., calling for violence), insult (e.g., making fun of appearance), irony, swearing, threat, other."*) but this was done as a post-hoc exercise and only for a small portion of the test set. It was not used in classification experiments, but only for separate analysis purposes.

(2016) investigate quality of comments and their use in summarisation. Wulczyn et al. (2017) investigate a related problem of detection of personal attacks and toxicity in user comments on Wikipedia articles, rather than news; and Zhang et al. (2018) also investigate Wikipedia comments from the point of view of detecting which conversations become toxic. While all of these may be useful for later work in WP3, none directly solve our problem here; additionally, all are limited to English data.

Table 1: Existing datasets for filtering user-generated comments on articles. Size is given in number of comments.

Corpus	Location	Domain	Language	Size	Type of annotation
Gazzetta	(Pavlopoulos et al., 2017a)	News	gr	1.6M	Moderation
SFU SOCC	(Kolhatkar et al., 2019)	News	en	663k	Constructiveness, toxicity
YNACC	(Napoles et al., 2017)	News	en	522k	Constructiveness, insults, off-topic
SENSEI	(Barker et al., 2016)	News	en	2k	Quality, tone, summaries
DETOX	(Wulczyn et al., 2017)	Wiki	en	115k	Personal attacks, aggression, toxicity
Zhang et al., 2018	(Zhang et al., 2018)	Wiki	en	7k	Personal attacks

2.3 Related work on offensive language detection

More resources are available for related specific tasks that correspond to particular phenomena or behaviours that moderators seek to block. In particular, recent years have seen much interest in the detection of offensive language and hate speech, mostly focusing on UGC in social media. Many public datasets have been created and distributed, many shared tasks have been run, and many classification systems developed and tested, although the exact definitions of the phenomena of interest vary with task and dataset – see Deliverable D3.1 for details. As an illustrative example, Waseem & Hovy (2016) define their *hate speech* category for Twitter as a message that:

1. *uses a sexist or racial slur;*
2. *attacks a minority;*
3. *seeks to silence a minority;*
4. *criticizes a minority (without a well founded argument);*
5. *promotes, but does not directly use, hatespeech or violent crime;*
6. *criticizes a minority and uses a straw man argument;*
7. *blatantly misrepresents truth or seeks to distort views on a minority with unfounded claims;*
8. *shows support of problematic hash tags. E.g.“#BanIslam”, “#whoriental”, “#whitegenocide”;*
9. *negatively stereotypes a minority;*
10. *defends xenophobia or sexism;*
11. *contains a screen name that is offensive, as per the previous criteria, the tweet is ambiguous (at best), and the tweet is on a topic that satisfies any of the above criteria.*

Most datasets are based on social media (mainly Twitter) posts, and most research in this area is monolingual, with English still the most popular language due to data availability (Wulczyn et al., 2017; Davidson et al., 2017). Most shared tasks organised on the topic of hate or offensive speech have been English-only (e.g. OffensEval (Zampieri et al., 2019)). Performance varies widely with dataset, domain and language. OffensEval 2019 reports maximum F1 score 0.829 on the offense classification task in English (Zampieri et al., 2019); de Gibert et al. (2018) report classification accuracy of 0.78 for white supremacy forum comments (again in English).

More recently, the focus has started to shift to other languages, with several shared tasks organised that cover other languages besides English, although still with a monolingual approach (e.g. EVALITA 2018 for Italian (Bai et al., 2018), GermEval 2018 for German (Wiegand et al., 2018), and Semeval 2019 task 5 including dataset partitions in Spanish and English (V. Basile et al., 2019)). One of the few multilingual hate speech studies was conducted by Ousidhoum et al. (2019), who tested a number of traditional bag-of-words and neural models on a multilingual dataset containing English, French and Arabic tweets that were manually labeled with six class hostility labels (abusive, hateful, offensive, disrespectful, fearful, normal). They report that multilingual models outperform monolingual models on some of the tasks.

However, the majority remain monolingual, and we are not aware of any data resources or tools in our primary target languages Croatian and Estonian.

2.4 Summary and Motivation

As this section has shown, no datasets or existing tools are currently available that directly provide resources for our needs here: automated news comment filtering in less-resourced languages, particularly in Croatian and Estonian. In order to develop and train automated classifiers, we need suitable datasets. However, for the exact domain of automatic moderation, the Gazzetta dataset of (Pavlopoulos et al., 2017b) is the only example from news, with the Wikipedia dataset of (Wulczyn et al., 2017) being quite closely related; but none are available in the languages required here. For specific subtasks such as offensive and hateful language, more is available, but most is monolingual and in English. Some multi-lingual work exists: Ousidhoum et al. (2019) present a multilingual hate speech study on English, French and Arabic tweets, and A. Basile & Rubagotti (2018) conduct cross-lingual experiments between Italian and English; again, this does not cover our languages or domain. Hatebase provides a highly multilingual collection of crowdsourced social media posts;[2] however, as its annotation is based only on submission by the public, and it contains no comparable non-abuse language, it is not currently suitable as training or evaluation data for a classifier of the kind needed here.

The closest match to our needs here are probably the Facebook dataset of socially unacceptable discourse in Slovenian of Ljubešić et al. (2019), and the Bulgarian news comment trolling data of Mihaylov & Nakov (2016), but neither are publicly available, neither are in the exact domain required, and neither include Croatian or Estonian. Our approach in Task T3.2 has therefore been to develop new classifiers, both training on the specific data we have (in the correct language and reflecting the moderation policy of the correct newspaper, although subject to real-world data constraints, Section 4), and separately training on standard datasets for the related task of offensive language detection in other languages (Section 3). We then combine these approaches via cross-lingual transfer (Section 5).

3 Monolingual UGC filtering and results

Given the focus of the user needs set out in Section 2.1, and the state of the art summarised in Section 2.2, our first steps in this task were to develop accurate classifiers for detection of offensive language and hate speech. Success in this task provides tools that can form the core of a comment filtering tool, and it is a task that can be approached using existing datasets.

3.1 Offensive language detection

The OffensEval public shared task at SemEval 2019 (Zampieri et al., 2019) provided a basis for offensive language detection, organising a competition on a dataset annotated for three sub-tasks: (A) identification of the presence of offensive language in a text; (B) automatic categorization of offense types; and (C) identification of the target of the offense. The dataset is composed of social media texts (Twitter), rather than news comments, but the length and informal nature of the text makes it a reasonable testing ground; however, the language is English, leaving the question of cross-lingual transfer open for later work (see Section 5).

The EMBEDDIA entry used a different model for each sub-task. For subtask A, we used the BERT model (Devlin et al., 2019) fine-tuned on the OffensEval dataset, while for subtasks B and C we developed a custom neural network architecture which combines bag-of-words features and automatically generated sequence-based features (see Figure 2). Results show that combining automatically and manually

[2] `http://hatebase.org/`

crafted features as an input to a neural architecture outperformed BERT transfer learning approach on quite imbalanced datasets and specific subtasks B and C (see Table 2).

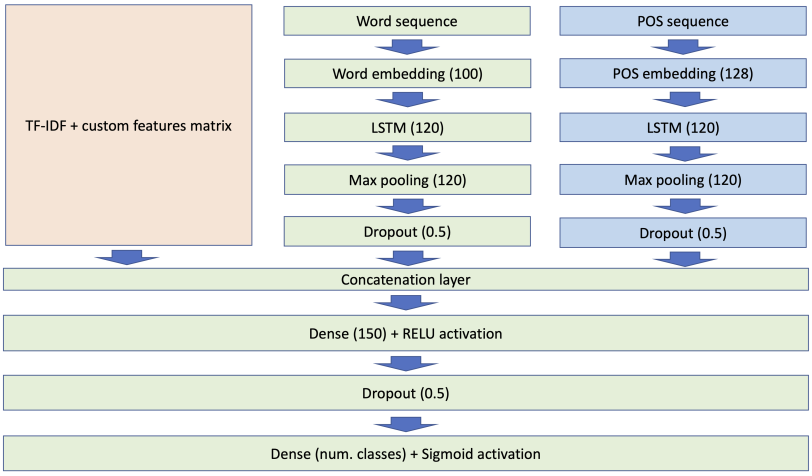


Figure 2: Network structure for subtasks B and C (Pelicon et al., 2019).

Sub-task	System	F1 (macro)	Accuracy
A	BERT	0.8078	0.8465
B	BOW+GloVeLSTM	0.6632	0.9042
C	BOW+GloVeLSTM+POS_LSTM	0.6133	0.7042

Table 2: Results on OffensEval 2019 subtasks A-C (Pelicon et al., 2019).

Results were competitive within the SemEval entries, achieving 4th place for Subtask A (the closest to the standard comment moderation task envisaged in the user needs story here), 18th for Subtask B and 5th for Subtask C (more specific tasks that may provide useful deeper information at a later stage). The architecture for Subtask A (the BERT model) is well suited for cross-lingual transfer with WP1's models (see Section 5 below); the model for Subtasks B and C requires features from target language data to give optimal performance, although in a cross-lingual setting these could be learned over time.

This work is described in full in (Pelicon et al., 2019), attached here as Appendix A.

3.2 Improving accuracy and robustness

In joint work with WP1 T1.4 in improvements of deep learning methods, we applied new adaptation of deep neural networks that can efficiently estimate prediction uncertainty by using Monte Carlo dropout regularization, which mimics Bayesian inference within neural networks. We applied this method to improve the robustness of classifiers trained to detect offensive language on three standard social media datasets: HatEval[3], YouToxic[4] and OffensiveTweets[5]. We compare the use of standard word embeddings from word2vec (Mikolov et al., 2013) and ELMo (Peters et al., 2018), and the use of sentence embeddings from the Universal Sentence Encoder (Cer et al., 2018), within a range of standard classifier approaches (logistic regression (LR), support vector machines (SVM) and a LSTM neural network).

[3] https://competitions.codalab.org/competitions/19935
[4] https://doi.org/10.5281/zenodo.2586669
[5] https://github.com/t-davidson/hate-speech-and-offensive-language

Table 3: Results using (a) word embeddings (b) sentence embeddings (Miok et al., 2019).

(a)

	HatEval			YouToxic			OffensiveTweets		
Model	**TF-IDF**	**W2V**	**ELMo**	**TF-IDF**	**W2V**	**ELMo**	**TF-IDF**	**W2V**	**ELMo**
LR	68.0 [2.4]	54.0 [13.6]	62.0 [6.8]	69.3 [3.0]	54.0 [3.0]	**76.6 [6.1]**	**77.2 [1.1]**	68.0 [2.4]	75.6 [1.2]
SVM	63.0 [5.1]	66.0 [3.7]	62.0 [12.9]	70.6 [4.2]	55.0 [3.4]	73.3 [5.5]	77.0 [0.7]	59.6 [1.5]	73.0 [1.9]
LSTM	69.0 [7.3]	67.0 [6.8]	66.0 [12.4]	66.6 [2.3]	59.3 [4.6]	74.3 [2.7]	73.4 [0.8]	75.0 [1.7]	74.7 [1.9]
MCD LSTM	67.0 [10.8]	**69.0 [6.6]**	67.0 [9.8]	66.0 [3.7]	59.3 [3.8]	75.3 [5.5]	71.1 [1.6]	72.0 [1.6]	75.2 [0.9]

(b)

	HatEval				YouToxic				OffensiveTweets			
Model	**Accuracy**	**Precision**	**Recall**	**F1**	**Accuracy**	**Precision**	**Recall**	**F1**	**Accuracy**	**Precision**	**Recall**	**F1**
LR	66.0 [12.4]	67.3 [15.3]	65.2 [15.9]	65.2 [13.1]	77.3 [4.1]	74.3 [7.3]	77.3 [3.6]	75.7 [5.3]	80.8 [1.0]	79.6 [1.9]	84.9 [1.2]	82.2 [1.1]
SVM	67.0 [12.1]	68.2 [15.2]	65.0 [15.8]	65.8 [13.3]	77.3 [6.2]	72.6 [8.6]	80.7 [7.4]	76.3 [7.6]	80.7 [1.3]	78.6 [2.0]	86.7 [1.0]	82.4 [1.2]
LSTM	70.0 [8.4]	70.8 [11.0]	63.1 [17.5]	66.2 [14.4]	76.6 [8.6]	73.4 [11.2]	79.2 [8.0]	75.8 [8.6]	80.7 [1.6]	82.8 [2.1]	79.7 [2.3]	81.1 [1.5]
MCD LSTM	**74.0 [10.7]**	73.4 [12.7]	78.4 [13.6]	74.9 [10.0]	**78.7 [5.8]**	74.7 [9.2]	80.9 [6.5]	77.5 [7.4]	**81.0 [1.2]**	81.5 [1.8]	82.5 [2.7]	81.9 [1.3]

We show that accuracy can be improved using sentence embeddings (see Table 3(b) vs. Table 3(a)) and also improved by the use of Monte Carlo Dropout (Gal & Ghahramani, 2016) to capture prediction uncertainty in regularization (see “MCD” vs. other results, Table 3). We also show that the reliability of the results can be visualized with a novel technique that helps to identify different types of errors and explain weaknesses in the classifier or wrongly labeled data - see Figure 3.

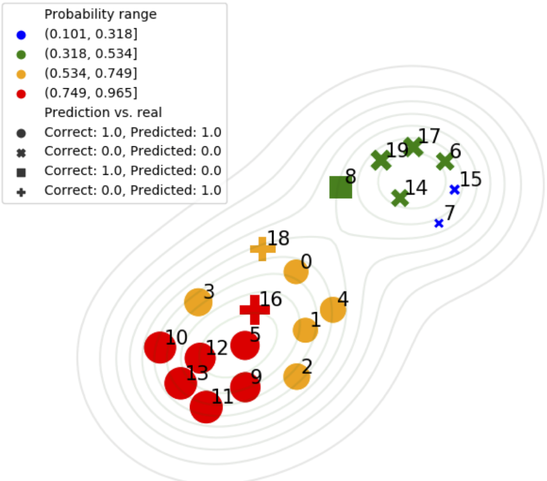


Figure 3: Example visualisation of individual predictions with probability ranges. Size of point corresponds to the neural network's mean probability for a given prediction. True positives are marked with circles, true negatives with crosses, false positives with squares, and false negatives as pluses (Miok et al., 2019).

This work is described in full in (Miok et al., 2019), attached here as Appendix B.

4 Monolingual news comment filtering and results

The work in Section 3 gives a sound basis for our overall classification approach, and shows it can deal with one of the primary language phenomena of interest in our task of news comment filtering (offensive language detection). However, the work so far was (a) monolingual, using standard datasets; (b) tested only in well-resourced languages, primarily English; and (c) restricted to social media, which although a type of UGC, is not entirely representative of the language encountered in news comments. In this section, we describe our work addressing issues (b) and (c) above: while staying with a monolingual approach, we transfer our classifiers to real news comment data in the less-resourced EMBEDDIA languages. Section 5 then addresses the issue (a).

For this, we use the new datasets collected and shared by our media partners Trikoder (TRI) and Ekspress Meedia (ExM), and described in Deliverable D3.1. This provides us with a large new source of data: over 60 million comments from the articles published online by three major news outlets in two less-resourced European languages:

- **24sata (`www.24sata.hr`):** The largest-circulation daily newspaper in Croatia, reaching on average 2 million readers daily.[6] Language: Croatian. Size: 21.5M comments.
- **Večernji List (`www.vecernji.hr`):** The third-largest daily newspaper in Croatia. Language: Croatian. Size: 9.6M comments.
- **Eesti Ekspress (`www.ekspress.ee`):** The largest weekly newspaper in Estonia, with a circulation of over 20,000. Languages: Estonian, Russian (articles are written in Estonian, but comments are often also in Russian). Size: 31.5M comments.

The disadvantage of the data collected this way is that the labelling is not explicitly collected by annotators to correspond to given language phenomena; instead, it is labelled implicitly with the actions of the in-house moderators. Comments that moderators decided should be blocked are recorded as such; in the case of 24sata and Večernji List (hereafter VL) they also record information about the reason for blocking, in terms of which rule in the newspaper's moderation policy was broken. These labels therefore correspond directly to the moderation behaviour desired; but can be extremely noisy, as moderators often make mistakes (in particular, missing comments that should be blocked), and blocking decisions are often subjective and arguable.

4.1 Supervised approach

The direct extension of the work so far, therefore, is to apply similar supervised neural network (NN) classifiers to the target data, using the moderators' decisions as target classification labels. We first trained and tested our models on a recent subset of the data from 2019, likely to have the least noisy labels of any portion. We compared the performance of a Naive Bayes (NB) classifier with NN models: a randomly-initialised LSTM, and two transfer-learning approaches using multilingual models trained on large standard datasets, LASER (Artetxe & Schwenk, 2019) and BERT (Devlin et al., 2019). Performance is reasonable in all cases, but does not reach the accuracies of the classifiers trained on standard datasets in Section 3 above - see Table 4.

Table 4: Classifier performance, as percentage accuracy. Columns are labelled ALL for all comments, BLK for positive instances only (blocked content), NON for negative instances only (non-blocked content).

	24sata			Večernji List			Ekspress		
Model	ALL	BLK	NON	ALL	BLK	NON	ALL	BLK	NON
NB	69.43	47.59	91.26	66.39	49.75	81.79	64.57	46.48	82.66
LSTM	71.52	61.70	81.33	65.39	54.47	75.50	63.02	41.96	84.09
LASER	70.74	70.11	71.36	63.31	59.77	66.59	61.58	47.07	76.10
mBERT	76.42	67.33	85.49	69.63	53.18	84.87	68.40	58.46	78.34

In the case of the 24sata and VL datasets, blocked comments are annotated with one of a wide range of reasons for blocking. This is based on a moderation policy which varies by newspaper: the policies for 24sata and VL are shown in Table 5 and Table 6, respectively. Comments should be blocked if they breach any of these rules, and implications for the comment author vary with the severity of the rule, from a minor warning to a permanent ban.

[6] `https://showcase.24sata.hr/2019_hosted_creatives/medijske-navike-hr-2019.pdf`

Table 5: Annotation schema for blocked comments, 24sata.

Rule ID	Description	Definition	Severity
1	Disallowed content	Advertising, content unrelated to the topic, spam, copyright infringement, citation of abusive comments or any other comments that are not allowed on the portal	Minor
2	Threats	Direct threats to other users, journalists, admins or subjects of articles, which may also result in criminal prosecution	Major
3	Hate speech	Verbal abuse, derogation and verbal attack based on national, racial, sexual or religious affiliation, hate speech and incitement	Major
4	Obscenity	Collecting and publishing personal information, uploading, distributing or publishing pornographic, obscene, immoral or illegal content and using a vulgar or offensive nickname that contains the name and surname of others	Major
5	Deception & trolling	Publishing false information for the purpose of deception or slander, and "trolling" - deliberately provoking other commentators	Minor
6	Vulgarity	Use of bad language, unless they are used as a stylistic expression, or are not addressed directly to someone	Minor
7	Language	Writing in other language besides Croatian, in other scripts besides Latin, or writing with all caps	Minor
8	Abuse	Verbally abusing of other users and their comments, article authors, and direct or indirect article subjects, calling the admins out or arguing with them in any way	Minor

Table 6: Annotation schema for blocked comments, Večernji List, together with corresponding Rule IDs from the 24sata schema (Table 5).

Rule ID	Corresponding 24sata rule ID(s)	Definition	Severity
1	3	Hate speech on a national, religious, sexual or any other basis	Major
2	2	Threats to other users, administrators, journalists or subjects of articles	Major
3	6, part 4, part 8	Insulting other users or use of bad language.	Minor
4	part 4	Publishing personal data	Minor
5	part 1, part 7	Chat, off-topic, writing in all caps, posting links	Minor
6	part 7	Writing in a script other than a Latin script	Minor
7	part 8	Challenging the administrators or arguing with then in any way	Minor
8	part 5	Posting false information	Minor
9	n/a	Using multiple user accounts	Permanent ban

The categories cover a broad range of grounds for moderation, and many categories potentially overlap: threats to others (rule 2); hate speech based on national, racial, sexual or religious affiliation (3); obscene or immoral content (4); bad language (6); and verbal abuse (8). However, they also include a range of other reasons: illegal content (rule 1); comments not allowed by the portal's rules (1); advertising (1); off-topic posts (1); copyright infringement (1); false information (5); use of language other than Croatian (7).

We next applied the same approach with a multi-class objective, to investigate the ability to detect different phenomena and potentially label moderated comments appropriately. Performance is good for some classifiers and some rules (see Table 7) but poor for the less common rules.

Table 7: Blocking rule classifier performance, measured as percentage accuracy, (a) 24sata (b) Večernji List.

	Model	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8
(a)	NB	43.01	0	3.23	0	5.67	5.97	8.77	2.74
	LSTM	62.42	0	56.05	0	50.52	75.37	43.86	57.53
	LASER	51.25	0	9.68	0	1.55	16.42	0	50.12
	mBERT	48.68	0	0	0	0	0	0	63.3

	Model	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8	Rule9
(b)	NB	6.61	5.47	4.56	0	6.4	100	4.55	0	33.73
	LSTM	25.73	20.64	33.65	50	35.22	0	13.64	0	40.41
	LASER	51.39	45.26	67.49	66.67	61.37	0	63.64	0	57.85
	mBERT	0	0	43.54	0	0	0	0	0	42.01

Performance seems highly dependent on variability in the dataset. When testing a model trained on the most recent 2019 data on a range of years, performance reaches good levels on similar 2019 data, reaching 62% macro-averaged F-score, with 81% accuracy on blocked comments. However, when tested on data from other years, performance drops noticeably, as moderator behaviour and/or data reliability changes - see Table 8.

Table 8: Binary classification performance over the yearwise testset using mBERT on 24sata dataset. Figures are shown as percentage accuracy overall and for the blocked and non-blocked content separately; as this experiment uses the full data for each year (rather than a balanced subset) we also give F_1 score macro-averaged over the two classes, and recall and precision for the blocked class only.

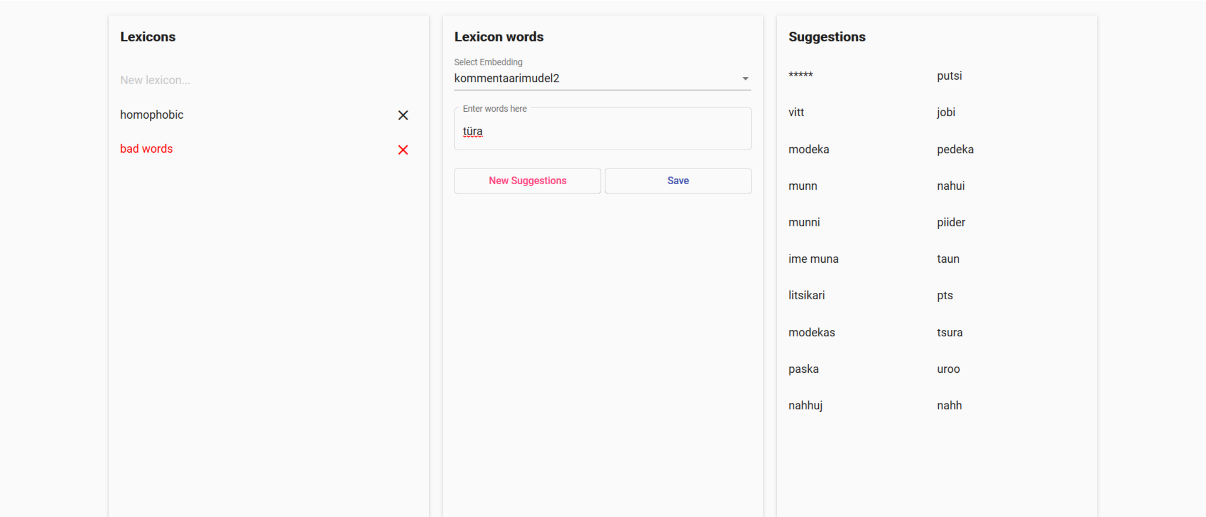
Year	Overall	Blocked	Non-blocked	F1-macro	Recall (BLK)	Precision (BLK)
2016	72.25	72.20	72.89	54.19	0.73	0.15
2017	75.17	76.16	64.84	58.10	0.65	0.21
2018	76.75	78.36	61.32	59.59	0.61	0.23
2019	80.03	81.19	67.32	62.07	0.67	0.25

We conclude that these approaches are suitable for cases where enough reliable data is available; further improvements are needed, which may come from one of two sources. One possibility is the use of human moderator intervention in a semi-supervised framework, and we investigate that in Section 4.2. Another is the use of cross-lingual transfer, bringing in information from more reliably curated datasets available in well-resourced languages, and using the new EMBEDDIA techniques from WP1 and WP2. We investigate that in Section 5.

This work is described in full in (Shekhar et al., 2020), attached here as Appendix C.

4.2 Semi-supervised approach

A direct way to overcome the lack of appropriate training data for moderating comments is to use a degree of manual intervention to increase the volume and quality of training data. To achieve this, we used a vocabulary-based approach via the TEXTA Toolkit (TTK), a component of the EMBEDDIA Media Assistant being developed in WP6. To construct specialised vocabularies for the specific filtering task, we used TTK's Lexicon Miner application to gather semantically similar or thematically related words (see Figure 4), based on the word2vec word embedding model (Mikolov et al., 2013) trained on the preprocessed and lemmatized ExM comment dataset. Each specialised vocabulary centered around a different theme, e.g. homophobia, racism, obscenity. We used these specialised vocabularies to extend the training data, by using TTK's Search application to find comments containing those vocabulary words.



türa

*****
vitt
modeka
munn
munni
ime muna
litsikari
modekas
paska
nahhuj

putsi
jobi
pedeka
nahui
piider
taun
pts
tsura
uroo
nahh

Figure 4: A view of the Lexicon Miner application in TTK.

Given the extended training data, we used Logistic Regression as a classifier with tf-idf weighted unigrams as features, and split the data 80-20 with the former used for training and hyperparameter tuning, and the latter for testing and computing scores. Optimal hyperparameters were chosen by using grid search with 5-fold cross-validation; precision, recall and F_1 scores were calculated on the test set. Two separate binary models were built (see Table 9) – the first model (*automatic_deletion*) is intended to classify comments that should be automatically banned without the need for doublechecking from an in-house moderator, and the second model (*manual_moderation*) is intended to classify borderline comments that require manual checking by the in-house moderator. Results (see Table 9) for both models are very promising, achieving F_1 scores of 0.9 or over.

Table 9: Results for the semi-supervised approach for comment moderation

Classifier	Model	F_1	Precision	Recall
Logistic Regression	automatic_deletion	0.92	0.98	0.87
	manual_moderation	0.90	0.99	0.82

5 Cross-lingual UGC filtering and results

Our next step was to investigate the use of cross-lingual transfer learning. The ability to train classifiers on datasets from well-resourced languages, and transfer them to less-resourced languages and/or tasks without much annotated data, is key for the overall objectives of EMBEDDIA: to produce tools which can be quickly adopted and used by news media companies on new datasets in different languages. Our experiments test the performance of cross-lingual models on the tasks of interest here: offensive language detection and comment filtering.

5.1 Applying standard models

Our first cross-lingual experiments used standard pre-trained embeddings: many such embeddings models are available, trained on a large number of languages using non-task-specific objectives (e.g. language modelling) on general unannotated language data.

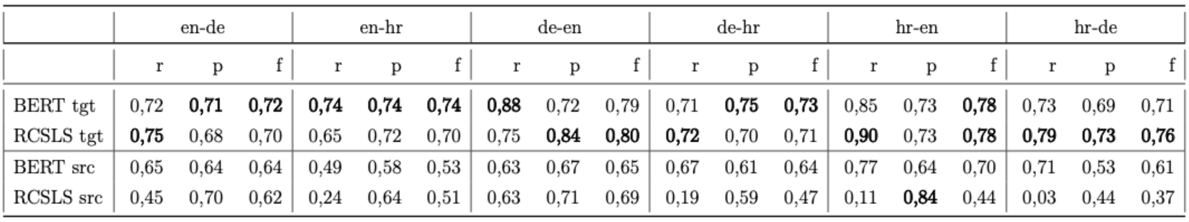
For these experiments we used three UGC datasets in different languages: for English, a collection of online forum comments labelled for hate speech content, sourced from the Stormfront website (de

Gibert et al., 2018); for German, a collection of hate speech targeting foreigners on public Facebook pages (Bretschneider & Peters, 2017); and for Croatian, a balanced subset of the EMBEDDIA news comment datasets from 24sata and VL described in Section 4. For the news comment dataset, as the transfer learning would be based on source datasets labelled specifically for hate speech (rather than other reasons for comment blocking, e.g. advertising), we used as positive examples only blocked comments which matched the “hate speech” rules in the moderation policies: rule 3 for 24sata (Table 5) and rule 1 for VL (Table 6).

We tested two cross-lingual transfer methods. First, we used a static (context-independent) embedding model, i.e. pre-trained fastText word embeddings (Bojanowski et al., 2017), available separately for each language, and aligned them with the RCSLS method (Joulin et al., 2018). We used these within a BiLSTM-CNN network architecture trained as a supervised classifier. Second, we used a contextual model, i.e. the pre-trained multilingual BERT (Devlin et al., 2019), fine-tuning the weights in the standard manner.

As Table 10 shows, fastText achieves reasonable performance in the cross-lingual setting in 2 out of 6 language pairs (English-to-German and German-to-English), but performs poorly on the other combinations. Multilingual BERT performs better, with F-scores over 0.6 in all but one pair. However, the drop from the monolingual setting is large, with absolute F-score reduction from 8% to 14%, and this is particularly acute for the key project language pairing English-to-Croatian, where cross-lingual performance is poor (F-score 0.53). We therefore conclude that better cross-lingual models are needed for the less-resourced, morphologically complex languages of interest on EMBEDDIA.

Table 10: Results using (a) word embeddings (b) sentence embeddings (Marinšek, 2019). ’tgt’ results are those achieved monolingually: training and testing on the target language data. ’src’ results are cross-lingual: training on source language data and testing on the target language.



	en-de			en-hr			de-en			de-hr			hr-en			hr-de		
	r	p	f	r	p	f	r	p	f	r	p	f	r	p	f	r	p	f
BERT tgt	0,72	**0,71**	**0,72**	**0,74**	**0,74**	**0,74**	**0,88**	0,72	0,79	0,71	**0,75**	**0,73**	0,85	0,73	**0,78**	0,73	0,69	0,71
RCSLS tgt	**0,75**	0,68	0,70	0,65	0,72	0,70	0,75	**0,84**	**0,80**	**0,72**	0,70	0,71	**0,90**	0,73	**0,78**	**0,79**	**0,73**	**0,76**
BERT src	0,65	0,64	0,64	0,49	0,58	0,53	0,63	0,67	0,65	0,67	0,61	0,64	0,77	0,64	0,70	0,71	0,53	0,61
RCSLS src	0,45	0,70	0,62	0,24	0,64	0,51	0,63	0,71	0,69	0,19	0,59	0,47	0,11	**0,84**	0,44	0,03	0,44	0,37

This work is described in full in (Marinšek, 2019), not attached here due to length, but available online from the University of Ljubljana repository at *https://repozitorij.uni-lj.si/IzpisGradiva.php?id=112851&lang=eng.*

5.2 Improving performance using EMBEDDIA models

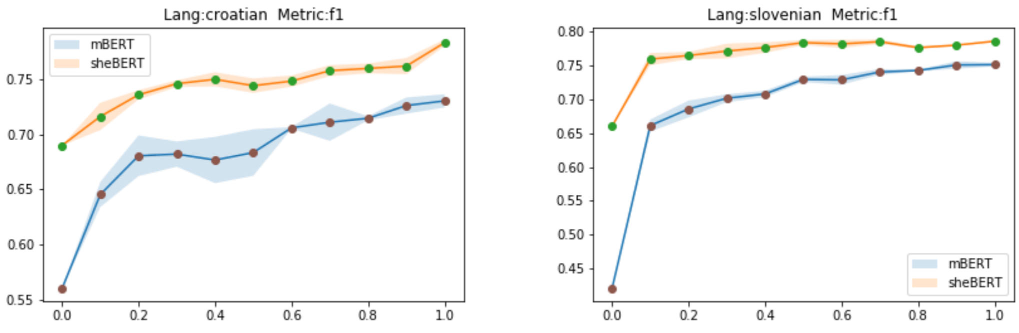
Our final step in this phase was to investigate the use of the new cross-lingual models developed for the EMBEDDIA languages in WP1 rather than the standard multilingual BERT used in the previous experiments. By using models more suitable for the target languages, we expect performance improvements, both in terms of overall classifier accuracy, and in terms of transfer between languages.

For this experiment, we used a combination of UGC data types, in order to simulate a likely practical use scenario: training on well-curated datasets labelled for known phenomena such as hate speech and offensive language, in well-resourced languages such as English; and testing on datasets in our target languages on UGC including news comments. We used the following datasets: standard datasets from shared tasks in English, Arabic and German, all taken from Twitter and labelled for offensive language (Zampieri et al., 2019; Mulki et al., 2019; Wiegand et al., 2018), a Slovenian language social media dataset taken from Facebook and labelled for offensive language (Ljubešić et al., 2019), and the EMBEDDIA 24sata news comment data described in previous sections, labelled by 24sata’s actual moderation process. For the news comment data, we took the subset of moderated comments that correspond to offensive language and hate speech classes (rules 2, 3, and 8 from Table 5). For our multi-lingual embedding model, we used the new CroSloEngual BERT, a tri-lingual model for English, Croatian and Slovenian developed in WP1 (Ulčar & Robnik-Šikonja, 2020).

As Figure 5 shows, the EMBEDDIA CroSloEngual BERT model significantly outperforms the standard multilingual BERT when the target language is one of the project languages, Croatian or Slovenian. Absolute improvements in F_1 score are 13% for Croatian and 24% for Slovenian in the fully cross-lingual case, where no target language training data is available, and pre-training is performed only on English data. Even in the fully trained case, where we assume that 100% of the target language training data is available, improvements are around 5% for Croatian, and 3% for Slovenian. Table 11 shows the benefit over the range of availability of target language training data, expressed as the Area Under the Curve (AUC) for the F1-score curves shown in Figure 5: CroSloEngual BERT gives noticeable improvements over mBERT for the less-resourced Croatian and Slovenian, but only very minor ones for English (where mBERT already provides good performance). As a sanity check, we notice that CroSloEngual BERT degrades performance for German and Arabic (as would be expected, for languages about which it has no information).

Table 11 also shows that cross-lingual transfer is more effective with CroSloEngual BERT, giving larger increases when either English (ENG) or Slovenian (SLO) source language training data is added to the target (TGT) language dataset. Figure 6 then shows the effectiveness of the model for cross-lingual transfer compared to a monolingual approach: when no target language training data is available, F_1 scores of 69% (Croatian) and 66% (Slovenian) are still achieved; performance improves further if target language training data can be procured, with the benefit of cross-lingual pre-training gradually diminishing as target language data volume grows.

Classifiers based on this approach have been implemented and supplied as dockerized software components for integration into WP6's Media Assistant - see Section 6.



(a) Croatian.

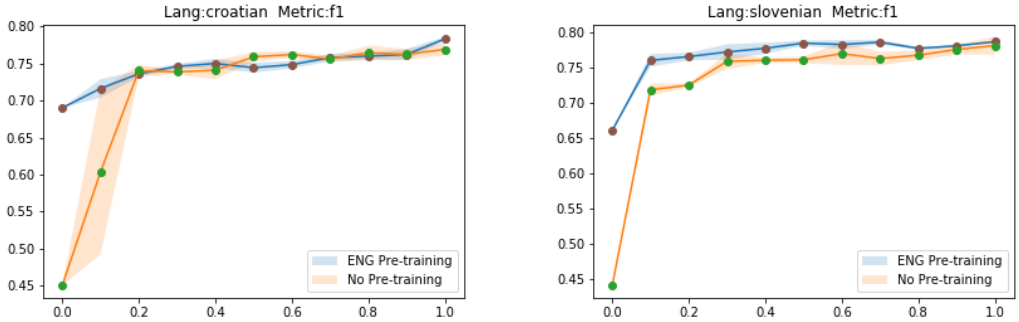
(b) Slovenian.

Figure 5: Classifier performance as macro-averaged F_1 score, for cross-lingual learning: classifiers are trained on a standard English offensive language dataset as an intermediate task, then tested on the target language. The x-axis shows the change as increasing amounts of target language data are added to the training. The two lines compare the standard multilingual BERT (labelled 'mBERT') (Devlin et al., 2019) vs the new EMBEDDIA CroSloEngual BERT (here labelled 'sheBERT') (Ulčar & Robnik-Šikonja, 2020) with varying amount of training data.

This work is described in full in (Pelicon et al., in preparation); in order to maintain anonymity for review, this is not attached here but can be made available on request and will be included as part of Deliverable D3.6.

Language	mBERT			CroSloEngual BERT		
	TGT	ENG+TGT	SLO+TGT	TGT	ENG+TGT	SLO+TGT
Croatian	68.15	68.72	69.13	70.43	73.45	73.48
English	73.31	-	74.22	73.65	-	74.94
Slovenian	68.93	70.34	-	73.89	76.41	-
German	68.47	67.87	69.19	65.23	64.14	62.81
Arabic	78.29	80.74	80.70	66.06	68.05	68.41

Table 11: Classifier performance across varying amounts of target language training dat, shown as Area Under the Curve (AUC) of F1-score.



(a) Croatian.

(b) Slovenian.

Figure 6: Classifier performance as macro-averaged F_1 score, showing the effect of cross-lingual learning using the EMBEDDIA CroSloEngual BERT model (Ulčar & Robnik-Šikonja, 2020). The two lines compare a monolingual approach (labelled ‘No pre-training’) vs pre-training on a standard English offensive language dataset as an intermediate task. The x-axis shows the change as increasing amounts of target language data are added to the training.

6 Associated outputs

The work described in this deliverable has resulted in the following resources:

Description	URL	Availability
Code for hate speech prediction	`github.com/EMBEDDIA/Hate-Speech-Prediction-Uncertainty`	Public (MIT)
Code and models for comment filtering	`github.com/EMBEDDIA/comment-filter`	Public (MIT)
Monolingual hate speech classification API	`classify.ijs.si/hate_speech/`	Online access
Multilingual hate speech classification API	`classify.ijs.si/ml_hate_speech/`	Online access

Parts of this work are also described in detail in the following publications, which are attached to this deliverable as appendices:

Citation	Status	Appendix
Pelicon, A., Martinc, M., & Kralj Novak, P. (2019, June). Embeddia at SemEval-2019 task 6: Detecting hate with neural network and transfer learning approaches. In Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval) (pp. 604–610). Minneapolis, Minnesota, USA: Association for Computational Linguistics.	Published	Appendix A
Miok, K., Nguyen-Doan, D., Škrlj, B., Zaharie, D., & Robnik-Šikonja, M. (2019). Prediction uncertainty estimation for hate speech classification. In International Conference on Statistical Language and Speech Processing (pp. 286–298). Springer.	Published	Appendix B
Marinšek, R. (2019). Cross-lingual embeddings for hate speech detection in comments. Master's thesis, University of Ljubljana, Faculty of Computer and Information Science. Available from `https://repozitorij.uni-lj.si/IzpisGradiva.php?id=112851&lang=eng`.	Published	(available online)
Shekhar, R., Pranjić, M., Pollak, S., Pelicon, A., & Purver, M. (2020). Automating news comment moderation with limited resources: Benchmarking in Croatian and Estonian. Journal of Language Technology and Computational Linguistics, to appear.	Accepted	Appendix C
Pelicon, A., Shekhar, R., Škrlj, B., Pollak, S., & Purver, M. (in preparation). Zero-shot cross-lingual content filtering: Offensive language and hate speech detection. Draft, in preparation.	Draft	(available on request)

7 Conclusions and further work

The objective of this task was to develop effective cross-lingual technologies for UGC, i.e. news comment filtering. As Section 5 shows, we have succeeded in developing classifiers for filtering news comments based on the presence of offensive language, that achieve good performance in a subset of the project languages, on real news comment data as well as social media data, even with no training data in the target language. We achieved this in several steps: first developing monolingual classifier methods for offensive language (Section 3); next applying the same techniques to news comment filtering (Section 4); then testing cross-lingual techniques (Section 5.1) and finally incorporating WP1's advances to improve these and achieve effective performance (Section 5.2). The classifiers developed have been implemented and integrated into the Media Assistant in WP6.

Next steps will extend the coverage of our classifiers to other project languages (e.g. building on the monolingual work in Estonian using a cross-lingual approach), and to a wider range of phenomena involved in comment filtering, e.g. spam, trolling and incitement behaviour. We will incorporate further advances from WP1, WP2 and T3.1 to improve performance and give more detailed information as the output. This will help in the report generation work in T3.3.

Bibliography

Artetxe, M., & Schwenk, H. (2019). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, *7*, 597–610.

Bai, X., Merenda, F., Zaghi, C., Caselli, T., & Nissim, M. (2018). RuG @ EVALITA: Hate speech detection in Italian social media. *EVALITA Evaluation of NLP and Speech Tools for Italian*, *12*, 245.

Barker, E., Paramita, M. L., Aker, A., Kurtic, E., Hepple, M., & Gaizauskas, R. (2016, September). The SENSEI annotated corpus: Human summaries of reader comment conversations in on-line news. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue* (pp. 42–52). Los Angeles: Association for Computational Linguistics.

Basile, A., & Rubagotti, C. (2018). CrotoneMilano for AMI at Evalita2018. A Performant, Cross-lingual Misogyny Detection System. In *EVALITA @ CLiC-it.*

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Pardo, F. M. R., ... 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* (pp. 54–63).

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Bretschneider, U., & Peters, R. (2017). Detecting offensive statements towards foreigners in social media. In *Proceedings of the 50th Hawaii International Conference on System Sciences.*

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R. S., ... Kurzweil, R. (2018). Universal sentence encoder. *CoRR*, *abs/1803.11175*. Retrieved from http://arxiv.org/abs/1803.11175

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In *Eleventh International AAAI Conference on Web and Social Media.*

de Gibert, O., Perez, N., García-Pablos, A., & Cuadros, M. (2018, October). Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)* (pp. 11–20). Brussels, Belgium: Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., & 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)* (pp. 4171–4186).

Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In *International Conference on Machine Learning* (p. 1050-1059).

Joulin, A., Bojanowski, P., Mikolov, T., Jégou, H., & Grave, E. (2018, October-November). Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 2979–2984). Brussels, Belgium: Association for Computational Linguistics.

Kolhatkar, V., Wu, H., Cavasso, L., Francis, E., Shukla, K., & Taboada, M. (2019, Nov 02). The SFU opinion and comments corpus: A corpus for the analysis of online news comments. *Corpus Pragmatics*.

Ljubešić, N., Fišer, D., & Erjavec, T. (2019). The FRENK datasets of socially unacceptable discourse in Slovene and English. *CoRR*, *abs/1906.02045*. Retrieved from http://arxiv.org/abs/1906.02045

Marinšek, R. (2019). *Cross-lingual embeddings for hate speech detection in comments* (Master's thesis, University of Ljubljana, Faculty of Computer and Information Science). Retrieved from https://repozitorij.uni-lj.si/IzpisGradiva.php?id=112851&lang=eng

Mihaylov, T., & Nakov, P. (2016, August). Hunting for troll comments in news community forums. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 399–405). Berlin, Germany: Association for Computational Linguistics.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems* (pp. 3111–3119).

Miok, K., Nguyen-Doan, D., Škrlj, B., Zaharie, D., & Robnik-Šikonja, M. (2019). Prediction uncertainty estimation for hate speech classification. In *International Conference on Statistical Language and Speech Processing* (pp. 286–298). Springer.

Mulki, H., Haddad, H., Ali, C. B., & Alshabani, H. (2019). L-HSAB: A Levantine Twitter dataset for hate speech and abusive language. In *Proceedings of the Third Workshop on Abusive Language Online* (pp. 111–118).

Napoles, C., Tetreault, J., Rosata, E., Provenzale, B., & Pappu, A. (2017, April). Finding good conversations online: The Yahoo news annotated comments corpus. In *Proceedings of the 11th linguistic annotation workshop* (pp. 13–23). Valencia, Spain: Association for Computational Linguistics.

Ousidhoum, N., Lin, Z., Zhang, H., Song, Y., & Yeung, D.-Y. (2019). Multilingual and multi-aspect hate speech analysis. *arXiv preprint arXiv:1908.11049*.

Pavlopoulos, J., Malakasiotis, P., & Androutsopoulos, I. (2017a, September). Deeper attention to abusive user content moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 1125–1135). Copenhagen, Denmark: Association for Computational Linguistics.

Pavlopoulos, J., Malakasiotis, P., & Androutsopoulos, I. (2017b, August). Deep learning for user comment moderation. In *Proceedings of the First Workshop on Abusive Language Online* (pp. 25–35). Vancouver, BC, Canada: Association for Computational Linguistics.

Pelicon, A., Martinc, M., & Kralj Novak, P. (2019, June). Embeddia at SemEval-2019 task 6: Detecting hate with neural network and transfer learning approaches. In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 604–610). Minneapolis, Minnesota, USA: Association for Computational Linguistics.

Pelicon, A., Shekhar, R., Škrlj, B., Pollak, S., & Purver, M. (in preparation). *Zero-shot cross-lingual content filtering: Offensive language and hate speech detection.* (Draft)

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 2227–2237).

Shekhar, R., Pranjić, M., Pollak, S., Pelicon, A., & Purver, M. (2020). Automating news comment moderation with limited resources: Benchmarking in Croatian and Estonian. *Journal of Language Technology and Computational Linguistics*, *to appear*.

Ulčar, M., & Robnik-Šikonja, M. (2020). FinEst BERT and CroSloEngual BERT: less is more in multilingual models. In *Proceedings of Text, Speech, and Dialogue, TSD 2020.* (accepted)

Waseem, Z., & Hovy, D. (2016, 01). Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop* (p. 88-93).

Wiegand, M., Siegel, M., & Ruppenhofer, J. (2018). Overview of the GermEval 2018 shared task on the identification of offensive language. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)* (pp. 1 – 10). Vienna, Austria: Austrian Academy of Sciences. Retrieved from `http://nbn-resolving.de/urn:nbn:de:bsz:mh39-84935`

Wulczyn, E., Thain, N., & Dixon, L. (2017). Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web* (pp. 1391–1399).

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019, June). SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 75–86). Minneapolis, Minnesota, USA: Association for Computational Linguistics.

Zhang, J., Chang, J., Danescu-Niculescu-Mizil, C., Dixon, L., Hua, Y., Taraborelli, D., & Thain, N. (2018, July). Conversations gone awry: Detecting early signs of conversational failure. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1350–1361). Melbourne, Australia: Association for Computational Linguistics.

Appendix A: Detecting hate with neural network and transfer learning approaches

Embeddia at SemEval-2019 Task 6: Detecting Hate with Neural Network and Transfer Learning Approaches

Andraž Pelicon, Matej Martinc, Petra Kralj Novak
Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
{Andraz.Pelicon, Matej.Martinc, Petra.Kralj.Novak}@ijs.si

Abstract

SemEval-2019 Task 6 was OffensEval: Identifying and Categorizing Offensive Language in Social Media. The task was further divided into three sub-tasks: offensive language identification, automatic categorization of offense types, and offense target identification. In this paper, we present the approaches used by the Embeddia team, who qualified as fourth, eighteenth and fifth on the three sub-tasks. A different model was trained for each sub-task. For the first sub-task, we used a BERT model fine-tuned on the provided dataset, while for the second and third tasks we developed a custom neural network architecture which combines bag-of-words features and automatically generated sequence-based features. Our results show that combining automatically and manually crafted features fed into a neural architecture outperform transfer learning approach on more unbalanced datasets.

1 Introduction

Over the years, computer-mediated communication, like the one on social media, has become one of the key ways people communicate and share opinions. Computer-mediated communication differs in many ways, both technically and culturally, from more traditional communication technologies (Kiesler et al., 1984). However, the ability to fully or partially hide our identity behind an internet persona leads people to type things they would never say to someone's face (Shaw, 2011). Not only is hate speech more likely to happen on the Internet, where anonymity is easily obtained and speakers are psychologically distant from their audience, but its online nature also gives it a far-reaching and determinative impact (Shaw, 2011). Although most forms of intolerance are not criminal, hate speech and other speech acts designed to harass and intimidate (rather than merely express criticism or dissent), deteriorate public discourse and opinions, which can lead to a more radicalized society.

Online communities, social media platforms, and technology companies have been investing heavily in ways to cope with offensive language to prevent abusive behavior in social media. Social media companies Facebook, Twitter and Google's YouTube have greatly accelerated their removal of online hate speech, and report reviewing over two-thirds of complaints within 24 hours. It has been proven in practice that naive word filtering systems do not manage to scale well to different forms of hate and aggression (Schmidt and Wiegand, 2017). The most promising strategy for detecting abusive language is to use advanced computational methods. This topic has attracted significant attention in recent years as evidenced in recent publications (Waseem et al., 2017; Davidson et al., 2017; Malmasi and Zampieri, 2018).

The SemEval-2019 Task 6 — OffensEval: Identifying and Categorizing Offensive Language in Social Media (Zampieri et al., 2019b) is to use machine learning text classification methods to identify offensive content and hate speech. The task organizers have provided a new dataset (Zampieri et al., 2019a) comprised of Twitter posts which employs a three-level hierarchical labeling scheme, according to the three hierarchically posed sub-tasks, where each sub-task serves as a stepping stone for the next sub-task. Sub-task A aims to identify offensive content, Sub-task B aims to classify offensive content as a targeted or untargeted offense, while Sub-task C aims to identify the target of the offense.

In this paper, we present the approaches used by the Embeddia team to tackle the three sub-tasks of SemEval-2019 Task 6: OffensEval. The Embeddia team qualified as fourth, eighteenth and fifth on Sub-tasks A, B and C, respectively. The Embed-

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 604–610
Minneapolis, Minnesota, USA, June 6–7, 2019. ©2019 Association for Computational Linguistics

dia team used different neural architectures and transfer learning techniques (Devlin et al., 2018). We also explore if combining automatically generated sequence-based features with more traditional manual feature engineering techniques improves the classification performance and how different classifiers perform on unbalanced datasets. Our results show that a combination of automatically and manually crafted features fed into a neural architecture outperforms the transfer learning approach on the more unbalanced datasets of Subtasks B and C.

This paper is organized as follows. In Section 2, we present related work in the area of offensive and hate speech detection. Section 3 describes in more detail the provided dataset and the methodology used for the task. Section 4 reviews the results we obtained on the three sub-tasks with our models. Section 5 concludes the paper and presents some ideas for future work.

2 Related Work

A number of workshops that dealt with offensive content, hate speech and aggression were organized in the past several years, which points to the increasing interest in the field. Due to important contributions of publications from TA-COS[1], Abusive Language Online[2], and TRAC[3], hate speech detection became better understood and established as a hard problem. The report on shared task from the TRAC workshop (Kumar et al., 2018) shows that of 45 systems trying to identify hateful content in English and Hindi Facebook posts, the best-performing ones achieved weighted macro-averaged F-scores of just over 0.6.

Schmidt and Wiegand (2017) note in their survey that supervised learning approaches are predominantly used for hate speech detection. Among those, the most widespread are support vector machines (SVM) and recurrent neural networks, which are emerging in recent times (Pavlopoulos et al., 2017). Zhang et al. (2018) devised a neural network architecture combining convolutional and gated recurrent layers for detecting hate speech, achieving state-of-the-art performance on several Twitter datasets. Malmasi and Zampieri (2018) used SVMs with different surface-level features, such as surface n-grams, word skip-grams and word representation n-grams induced with Brown clustering. They concluded that surface n-grams perform well for hate speech detection but also noted that these features might not be enough to discriminate between profanity and hate speech with high accuracy and that deeper linguistic features might be required for this scenario.

A common difficulty that arises with supervised approaches for hate speech and aggression detection is a skewed class distribution in datasets. Davidson et al. (2017) note that in the dataset used in the study only 5% of tweets were labeled as hate speech. To counteract this, datasets are often resampled with different techniques to improve on the predictive power of the systems over all classes. Aroyehun and Gelbukh (2018) increased the size of the used dataset by translating examples to four different languages, namely French, Spanish, German, and Hindi, and translating them back to English. Their system placed first in the Aggression Detection in Social Media Shared Task of the aforementioned TRAC workshop.

A recently emerging technique in the field of natural language processing (NLP) is the employment of transfer learning (Howard and Ruder, 2018; Devlin et al., 2018). The main idea of these approaches is to pretrain a neural language model on large general corpora and then fine-tune this model for a task at hand by adding an additional task-specific layer on top of the language model and train it for a couple of additional epochs. A recent model called Bidirectional encoder representations from transformers (BERT) (Devlin et al., 2018) was pretrained on the concatenation of BooksCorpus (800M words) (Zhu et al., 2015) and English Wikipedia (2,500M words) and then successfully applied to a number of NLP tasks without changing its core architecture and with relatively inexpensive fine-tuning for each specific task. According to our knowledge, it has not been applied on a hate speech detection task yet, however it reached state-of-the-art results in the question answering task on the SQuAD dataset (Rajpurkar et al., 2016) as well as beat the baseline models in several language inference tasks.

3 Methodology and Data

This section describes the tasks, the dataset, the methodology used and the experiments.

[1] http://ta-cos.org/

[2] https://sites.google.com/site/abusivelanguageworkshop2017/

[3] https://sites.google.com/view/trac1/home

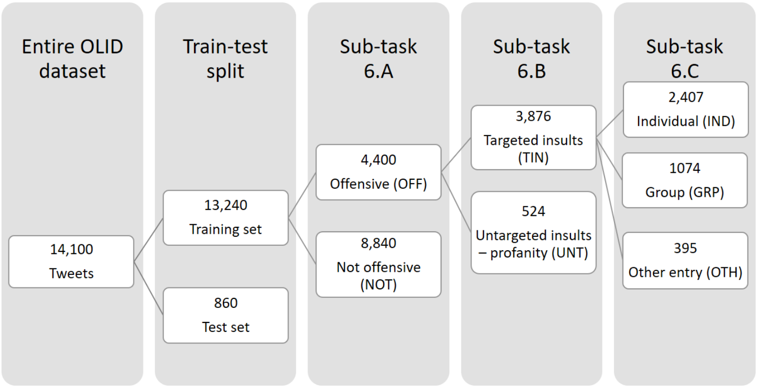


Figure 1: Schema of SemEval-Task 6: OffensEval: Identifying and Categorizing Offensive Language in Social Media. The hierarchy of the sub-tasks and respective dataset sizes.

3.1 Dataset

The SemEval-2019 Shared Task 6: Identifying and Categorizing Offensive Language in Social Media was divided into three sub-tasks, namely offensive language identification (Sub-task A), automatic categorization of offense types (Sub-task B) and offense target identification (Sub-task C). The organizers provided a new dataset called OLID (Zampieri et al., 2019a) which includes tweets labeled according to the three-level hierarchical model. On the very first level, each tweet is labeled as offensive (OFF) or not offensive (NOT). All the offensive tweets are then labeled as targeted insults (TIN) or as untargeted insults (UNT), which simply contain profanity. On the last level, all targeted insults are categorized as targeting an individual (IND), a group (GRP) or other entity (OTH). The dataset contains 14,100 tweets split into training and test sets. The training set containing 13,240 tweets and the test set without labels were made available to the participants for the task. The inspection of the dataset reveals that the classes at first level are slightly imbalanced with the imbalances between classes getting more prominent with each subsequent level. A more detailed breakdown of the dataset is presented in Figure 1. We didn't use any additional datasets in any of the three sub-tasks.

3.2 Methodology

According to the findings from the related work, we decided to test two different types of architectures. First was a pretrained BERT model, which was fine-tuned on the provided dataset for distinguishing offensive and not offensive posts in the Sub-task A. For the sub-tasks B and C, a neural network architecture was developed, which tried to achieve synergy between two types of features that both proved successful in the past approaches to the task at hand, by basing its predictions on a combination of classical bag-of-words features and automatically generated sequence-based features. The three models, as well as their source code, are available for download in a public repository[4].

Three models were trained using the provided dataset, one for each sub-task. In the Sub-task A, the large pretrained BERT transformer with 24 layers of size 1024 and 16 self-attention heads was used for generating predictions on the official test set. A linear sequence classification head responsible for producing final predictions was added on top of the pretrained language model and the whole classification model was fine-tuned on the SemEval input data for 3 epochs. For training, a batch size of 8 and a learning rate of 2e-5 were used. The training dataset for the Sub-task A was randomly split into a training set containing 80% of the tweets and a validation set containing 20% of the tweets. Only a small amount of text preprocessing was needed on the data for the Sub-task A since the dataset already had all Twitter user mentions replaced by @USER tokens and all URLs by URL tokens. Additionally, we lowercased and tokenized the tweets using BERT's built-in tokenizer.

For Sub-task B, the non-offensive tweets were first filtered out of the original dataset. The re-

[4] https://gitlab.com/Andrazp/embeddia-semeval2019

duced dataset had 4400 tweets. To offset the lower quantity of data, we decided to split the dataset into a training set containing 90% of the data and a validation set containing 10% of the data. The second issue with the data was a severe class imbalance as only 12% of tweets in the filtered dataset were labeled as untargeted insults. We decided to resample the dataset in order to minimize the impact of the imbalance on our training. The approach that yielded the best results based on the validation set performance was to randomly remove the instances of the majority class until the classes were balanced. The remaining instances were lowercased and tokenized with the tweet tokenizer from the NLTK package (Bird et al., 2009). Stopwords were also removed from every tweet using an English stopwords list provided in the NLTK package.

As the BERT model was showing worse performance on the resampled data according to the validation set results, a new neural network architecture was devised for this sub-task (Figure 2). The neural architecture takes two inputs. The first input is a term frequency-inverse document frequency (tf-idf) weighted bag-of-words matrix calculated on 1- to 5-grams and character 1- to 7- grams using sublinear term frequency scaling. N-grams with document frequencies less than 5 were removed from the final matrix. Furthermore, the following additional features are generated for each tweet in the training set and added to the tf-idf matrix:

- The number of insults: using a list of English insults,[5] the insults in each tweet are counted and their number is added to the matrix as a feature.
- The length of the longest punctuation sequence: for every punctuation mark that appears in the Python built-in list of punctuations, its longest sequence is found in each tweet. The length of the sequence is then added as a feature.
- Sentiment of the tweets: the sentiment of each tweet is predicted by an SVM model (Mozetič et al., 2016) pretrained on English tweets. The model classifies each tweet as positive, neutral or negative. The predictions are then encoded and added as features.

The second input is word sequences, which are fed into an embedding layer with pretrained 100-dimensional GloVe (Pennington et al., 2014) embedding weights trained on a corpus of English tweets. The pretrained embeddings are additionally fine-tuned during the training process on the dataset for the task. The resulting embeddings are fed to an LSTM layer with 120 units, on the output of which we perform global max pooling. We perform a dropout operation on the max pooling output and the resulting vectors are concatenated with the tf-idf vectors. The resulting concatenation is sent to a fully-connected hidden layer with 150 units, the output of which is fed to a rectified linear unit (RELU) activation function. After performing dropout, final predictions are produced by a fully-connected hidden layer with a sigmoid activation function. For training, we use a batch size of 16 and Adam optimizer with a learning rate of 0.001. We trained the model for a maximum of 10 epochs and validated its performance on the validation set after every epoch. The best performing model was later used for generating predictions on the official test set.

For Sub-task C, the dataset was additionally filtered by removing the tweets that were labeled as non-targeted insults. The class imbalance for this task was even more prominent with only 28% of tweets being labeled as insults targeted towards groups and 10% as targeted insults that do not target an individual or a specific group of people. In light of such class imbalance, the dataset was again undersampled by removing 75% of tweets from the majority class and 50% percent of tweets from the middle class. Due to the dataset being even more aggressively filtered, the 90-10% split from the previous sub-task was kept. A modified version of the neural architecture from Sub-task B was used for prediction. We tried to capture the relationship between insults and their targets using sentence structure information. To this end, we added a third input to the neural architecture that accepts sequences of part-of-speech (POS) tags. First, all the tweets were POS-tagged using the POS tagger from the NLTK package and the resulting POS tag sequences were then fed to a randomly initialized embedding layer. Output embeddings are then fed to an LSTM layer with 120 units, on the output of which we performed global

[5] http://metadataconsulting.blogspot.com/2018/09/Google-Facebook-Office-365-Dark-Souls-Bad-Offensive-Profanity-key-word-List-2648-words.html

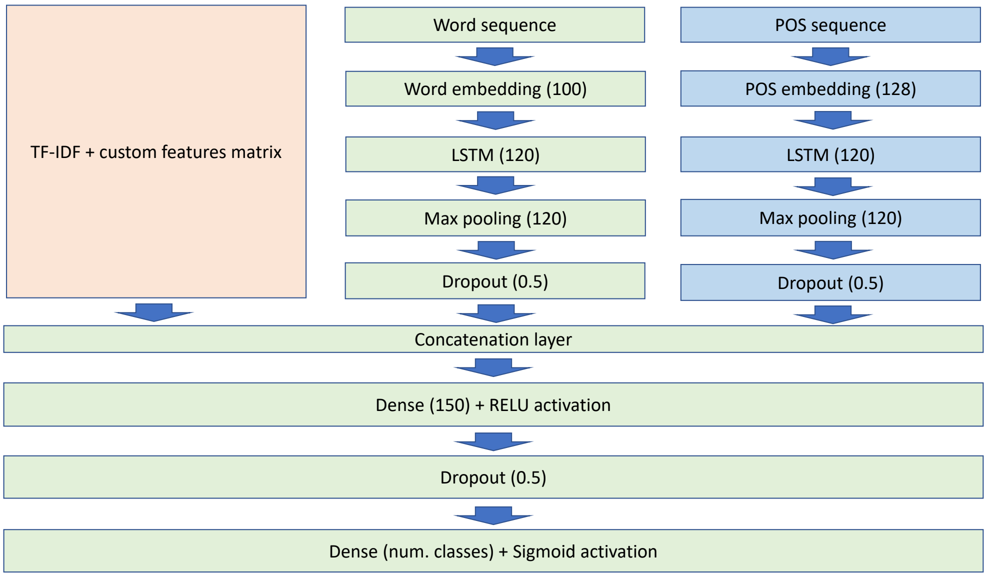


Figure 2: System architecture used in Sub-tasks B and C. Parts of the infrastructure depicted in blue were only used in Sub-task C.

max pooling. Next, dropout was applied, and the resulting vector matrix was then concatenated with the matrices from other inputs and sent to the fully-connected layer (see Figure 2).

4 Results

The results on the official test sets for all three tasks are presented in Table 1. In the Sub-task A, our BERT model, fine-tuned on the provided dataset, achieved a macro-averaged F1 score of 0.808. When we compare this result to other teams participating in the SemEval-2019 OffensEval Sub-task A, we rank fourth.

As the dataset was filtered and the class imbalances became more prominent in the subsequent tasks, the performance of our models started to deteriorate. Even though the undersampling of the dataset to offset class imbalances further reduced the available data, it proved to be the best way to ensure somewhat reliable predictions. The models for Sub-task B and C had macro-averaged F1 scores of 0.663 and 0.613 respectively and placed eighteenth and fifth overall in the SemEval-2019 OffensEval official ranking.

A closer look at the confusion matrices further confirms our claim about the impact of class imbalances on our systems' performance. While the predictions for both classes were fairly accurate in the Sub-task A (Figure 3a), we can see a dwindling performance on the untargeted insults (UNT) class in Sub-task B (Figure 3b) where approximately two thirds of the instances were misclassified as targeted insults (TIN) class on the test set.

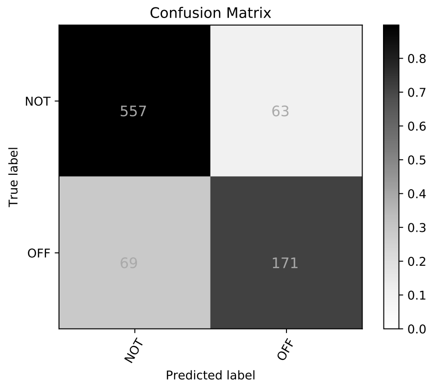
The confusion matrix for Sub-task C (Figure 3c) paints a very similar picture. Even though the majority individual (IND) and middle group (GRP) classes were heavily imbalanced in the original dataset, our model was still able to successfully discriminate between them. However, it again performed subpar on the minority other entity (OTH) class, which was heavily underrepresented compared to the other two. Of the 35 instances in the test set, three out of four were misclassified.

5 Conclusion

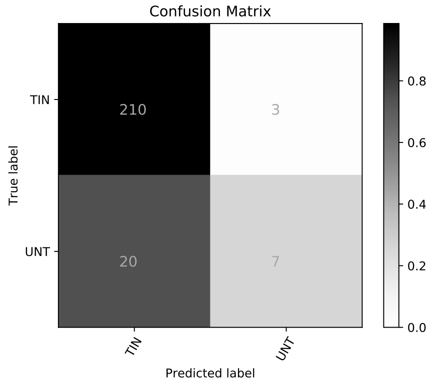
In this paper, we presented the results of the Embeddia team on the SemEval-2019 Task 6: OffensEval: Identifying and Categorizing Offensive Language in Social Media using the dataset provided by the organizers of the task. The task was further divided into three sub-tasks, namely offensive language identification (Sub-task A), automatic categorization of offense types (Sub-task B) and offense target identification (Sub-task C). We trained three models, one for each sub-task. For Sub-task A, we used a BERT model fine-tuned on the OLID dataset, while for the second and third tasks we developed a neural network architecture

Sub-task	System	F1 (macro)	Accuracy
A	BERT	0.8078	0.8465
B	BOW+GloVeLSTM	0.6632	0.9042
C	BOW+GloVeLSTM+POS_LSTM	0.6133	0.7042

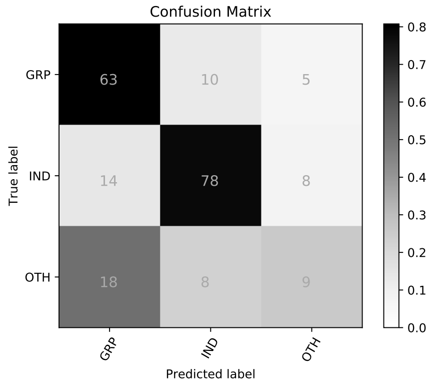
Table 1: Results of the submitted systems for each sub-task.



(a) Confusion matrix for the BERT system, fine-tuned on the provided dataset for Sub-task A.



(b) Confuaion matrix of the two-input neural network with a LSTM based on word sequences and a bag-of-words matrix for Sub-task B.



(c) Confusion matrix of the three-input neural network with an LSTM based on word sequences, LSTM based on part-of-speech tags sequences and a bag-of-words matrix for Sub-task C.

which combines bag-of-words features and automatically generated sequence-based features. Our models ranked fourth, eighteenth and fifth in Subtasks A, B and C, respectively.

We noticed that the class imbalances in the datasets had a significant impact on the performance of our systems and were especially deteriorating for the performance of the BERT system. To counteract the impact of class imbalances we used various techniques to resample the original datasets. While randomly removing instances from the majority classes proved to be the most consistent approach to improve the predictive power of our systems, the effect of the class imbalance persisted.

Our aim for the future is to make the systems more robust to imbalanced data to better generalize over all the classes. Since we already have several models that perform adequately, a good next step would be to implement an ensemble model using a plurality voting or a gradient boosting scheme. We will also conduct an ablation study to identify which features work particularly well for offensive content and hate speech detection.

Acknowledgments

This paper is supported also by the European Unions Horizon 2020 research and innovation programme under Grant No. 825153, EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media). The results of this publication reflect only the authors views and the Commission is not responsible for any use that may be made of the information it contains. The authors acknowledge also the financial support from the Slovenian Research Agency core research programme Knowledge Technologies (P2-0103). The Titan Xp used for this research was donated by the NVIDIA Corporation.

References

Segun Taofeek Aroyehun and Alexander Gelbukh. 2018. Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 90–97.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of ICWSM*.

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*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Sara Kiesler, Jane Siegel, and Timothy W McGuire. 1984. Social psychological aspects of computer-mediated communication. *American psychologist*, 39(10):1123.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking Aggression Identification in Social Media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbulling (TRAC)*, Santa Fe, USA.

Shervin Malmasi and Marcos Zampieri. 2018. Challenges in Discriminating Profanity from Hate Speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30:1–16.

Igor Mozetič, Miha Grčar, and Jasmina Smailović. 2016. Multilingual twitter sentiment classification: The role of human annotators. *PloS one*, 11(5):e0155036.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1125–1135.

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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Anna Schmidt and Michael Wiegand. 2017. A Survey on Hate Speech Detection Using Natural Language Processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media. Association for Computational Linguistics*, pages 1–10, Valencia, Spain.

LaShel Shaw. 2011. Hate speech in cyberspace: bitterness without boundaries. *Notre Dame JL Ethics & Pub. Pol'y*, 25:279.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding Abuse: A Typology of Abusive Language Detection Subtasks. In *Proceedings of the First Workshop on Abusive Langauge Online*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of NAACL*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval)*.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In *Lecture Notes in Computer Science*. Springer Verlag.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

Appendix B: Prediction uncertainty estimation for hate speech classification

Prediction Uncertainty Estimation for Hate Speech Classification

Kristian Miok[1](✉), Dong Nguyen-Doan[1], Blaž Škrlj[2], Daniela Zaharie[1], and Marko Robnik-Šikonja[3]

[1] Computer Science Department, West University of Timisoara,
Bulevardul Vasile Pârvan 4, 300223 Timişoara, Romania
{kristian.miok,dong.nguyen10,daniela.zaharie}@e-uvt.ro

[2] Jožef Stefan Institute and Jožef Stefan International Postgraduate School,
Jamova 39, 1000 Ljubljana, Slovenia
blaz.skrlj@ijs.si

[3] Faculty of Computer and Information Science, University of Ljubljana,
Večna pot 113, 1000 Ljubljana, Slovenia
marko.robnik@fri.uni-lj.si

Abstract. As a result of social network popularity, in recent years, hate speech phenomenon has significantly increased. Due to its harmful effect on minority groups as well as on large communities, there is a pressing need for hate speech detection and filtering. However, automatic approaches shall not jeopardize free speech, so they shall accompany their decisions with explanations and assessment of uncertainty. Thus, there is a need for predictive machine learning models that not only detect hate speech but also help users understand when texts cross the line and become unacceptable.

The reliability of predictions is usually not addressed in text classification. We fill this gap by proposing the adaptation of deep neural networks that can efficiently estimate prediction uncertainty. To reliably detect hate speech, we use Monte Carlo dropout regularization, which mimics Bayesian inference within neural networks. We evaluate our approach using different text embedding methods. We visualize the reliability of results with a novel technique that aids in understanding the classification reliability and errors.

Keywords: Prediction uncertainty estimation · Hate speech classification · Monte Carlo dropout method · Visualization of classification errors

1 Introduction

Hate speech represents written or oral communication that in any way discredits a person or a group based on characteristics such as race, color, ethnicity, gender, sexual orientation, nationality, or religion [35]. Hate speech targets disadvantaged social groups and harms them both directly and indirectly [33]. Social

C. Martín-Vide et al. (Eds.): SLSP 2019, LNAI 11816, pp. 286–298, 2019.
https://doi.org/10.1007/978-3-030-31372-2_24

networks like Twitter and Facebook, where hate speech frequently occurs, receive many critics for not doing enough to deal with it. As the connection between hate speech and the actual hate crimes is high [4], the importance of detecting and managing hate speech is not questionable. Early identification of users who promote such kind of communication can prevent an escalation from speech to action. However, automatic hate speech detection is difficult, especially when the text does not contain explicit hate speech keywords. Lexical detection methods tend to have low precision because, during classification, they do not take into account the contextual information those messages carry [11]. Recently, contextual word and sentence embedding methods capture semantic and syntactic relation among the words and improve prediction accuracy.

Recent works on combining probabilistic Bayesian inference and neural network methodology attracted much attention in the scientific community [23]. The main reason is the ability of probabilistic neural networks to quantify trustworthiness of predicted results. This information can be important, especially in tasks were decision making plays an important role [22]. The areas which can significantly benefit from prediction uncertainty estimation are text classification tasks which trigger specific actions. Hate speech detection is an example of a task where reliable results are needed to remove harmful contents and possibly ban malicious users without preventing the freedom of speech. In order to assess the uncertainty of the predicted values, the neural networks require a Bayesian framework. On the other hand, Srivastava et al. [32] proposed a regularization approach, called dropout, which has a considerable impact on the generalization ability of neural networks. The approach drops some randomly selected nodes from the neural network during the training process. Dropout increases the robustness of networks and prevents overfitting. Different variants of dropout improved classification results in various areas [1]. Gal and Ghahramani [14] exploited the interpretation of dropout as a Bayesian approximation and proposed a Monte Carlo dropout (MCD) approach to estimate the prediction uncertainty. In this paper, we analyze the applicability of Monte Carlo dropout in assessing the predictive uncertainty.

Our main goal is to accurately and reliably classify different forms of text as hate or non-hate speech, giving a probabilistic assessment of the prediction uncertainty in a comprehensible visual form. We also investigate the ability of deep neural network methods to provide good prediction accuracy on small textual data sets. The outline of the proposed methodology is presented in Fig. 1.

Our main contributions are:

- investigation of prediction uncertainty assessment to the area of text classification,
- implementation of hate speech detection with reliability output,
- evaluation of different contextual embedding approaches in the area of hate speech,
- a novel visualization of prediction uncertainty and errors of classification models.

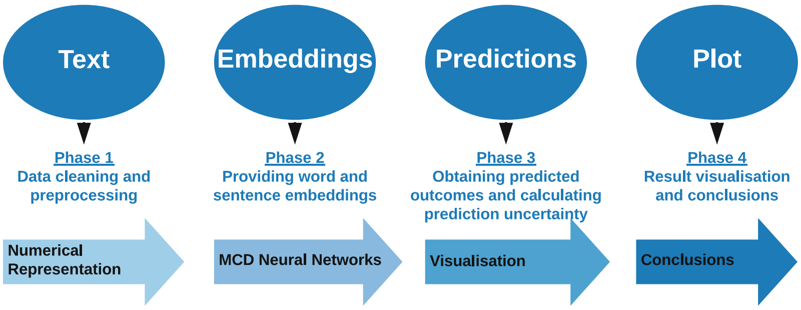

Text
Embeddings
Predictions
Plot
Phase 1
Data cleaning and preprocessing
Phase 2
Providing word and sentence embeddings
Phase 3
Obtaining predicted outcomes and calculating prediction uncertainty
Phase 4
Result visualisation and conclusions
Numerical Representation
MCD Neural Networks
Visualisation
Conclusions

Fig. 1. The diagram of the proposed methodology.

The paper consists of six sections. In Sect. 2, we present related works on hate speech detection, prediction uncertainty assessment in text classification context, and visualization of uncertainty. In Sect. 3, we propose the methodology for uncertainty assessment using dropout within neural network models, as well as our novel visualization of prediction uncertainty. Section 4 presents the data sets and the experimental scenario. We discuss the obtained results in Sect. 5 and present conclusions and ideas for further work in Sect. 6.

2 Related Work

We shortly present the related work in three areas which constitute the core of our approach: hate speech detection, recurrent neural networks with Monte Carlo dropout for assessment of prediction uncertainty in text classification, and visualization of predictive uncertainty.

2.1 Hate Speech Detection

Techniques used for hate speech detection are mostly based on supervised learning. The most frequently used classifier is the Support Vector Machines (SVM) method [30]. Recently, deep neural networks, especially recurrent neural network language models [20], became very popular. Recent studies compare (deep) neural networks [9,12,28] with the classical machine learning methods.

Our experiments investigate embeddings and neural network architectures that can achieve superior predictive performance to SVM or logistic regression models. More specifically, our interest is to explore the performance of MCD neural networks applied to the hate speech detection task.

2.2 Prediction Uncertainty in Text Classification

Recurrent neural networks (RNNs) are a popular choice in text mining. The dropout technique was first introduced to RNNs in 2013 [34] but further research revealed negative impact of dropout in RNNs, especially within language modeling. For example, the dropout in RNNs employed on a handwriting recognition task, disrupted the ability of recurrent layers to effectively model sequences [25]. The dropout was successfully applied to language modeling by [36] who applied it only on fully connected layers. The then state-of-the-art results were explained with the fact that by using the dropout, much deeper neural networks can be constructed without danger of overfitting. Gal and Ghahramani [15] implemented the variational inference based dropout which can also regularize recurrent layers. Additionally, they provide a solution for dropout within word embeddings. The method mimics Bayesian inference by combining probabilistic parameter interpretation and deep RNNs. Authors introduce the idea of augmenting probabilistic RNN models with the prediction uncertainty estimation. Recent works further investigate how to estimate prediction uncertainty within different data frameworks using RNNs [37]. Some of the first investigation of probabilistic properties of SVM prediction is described in the work of Platt [26]. Also, investigation how Bayes by Backprop (BBB) method can be applied to RNNs was done by [13].

Our work combines the existing MCD methodology with the latest contextual embedding techniques and applies them to hate speech classification task. The aim is to obtain high quality predictions coupled with reliability scores as means to understand the circumstances of hate speech.

2.3 Prediction Uncertainty Visualization in Text Classification

Visualizations help humans in making decisions, e.g., select a driving route, evacuate before a hurricane strikes, or identify optimal methods for allocating business resources. One of the first attempts to obtain and visualize latent space of predicted outcomes was the work of Berger et al. [2]. Prediction values were also visualized in geo-spatial research on hurricane tracks [10,29]. Importance of visualization for prediction uncertainty estimation in the context of decision making was discussed in [17,18].

We are not aware of any work on prediction uncertainty visualization for text classification or hate speech detection. We present visualization of tweets in a two dimensional latent space that can reveal relationship between analyzed texts.

3 Deep Learning with Uncertainty Assessment

Deep learning received significant attention in both NLP and other machine learning applications. However, standard deep neural networks do not provide information on reliability of predictions. Bayesian neural network (BNN) methodology can overcome this issue by probabilistic interpretation of model parameters. Apart from prediction uncertainty estimation, BNNs offer robustness to overfitting and can be efficiently trained on small data sets [16]. However, neural networks that apply Bayesian inference can be computationally expensive, especially the ones with the complex, deep architectures. Our work is based on Monte Carlo Dropout (MCD) method proposed by [14]. The idea of this approach is to capture prediction uncertainty using the dropout as a regularization technique.

In contrast to classical RNNs, Long Short-term Memory (LSTM) neural networks introduce additional gates within the neural units. There are two sources of information for specific instance t that flows through all the gates: input values x_t and recurrent values that come from the previous instance h_{t-1}. Initial attempts to introduce dropout within the recurrent connections were not successful, reporting that dropout brakes the correlation among the input values. Gal and Ghahramani [15] solve this issue using predefined dropout mask which is the same at each time step. This opens the possibility to perform dropout during each forward pass through the LSTM network, estimating the whole distribution for each of the parameters. Parameters' posterior distributions that are approximated with such a network structure, $q(\omega)$, is used in constructing posterior predictive distribution of new instances y^*:

$$p(y^*|x^*, D) \approx \int p\big(y^*|f^{\omega}(x^*)\big)\ q(\omega)d\omega, \tag{1}$$

where $p\big(y^*|f^{\omega}(x^*)\big)$ denotes the likelihood function. In the regression tasks, this probability is summarized by reporting the means and standard deviations while for classification tasks the mean probability is calculated as:

$$\frac{1}{K}\sum_{k=1}^{K} p(y^*|x^*, \hat{\omega}_k) \tag{2}$$

where $\hat{\omega}_k \sim q(\omega)$. Thus, collecting information in K dropout passes throughout the network during the training phase is used in the testing phase to generate (sample) K predicted values for each of the test instance. The benefit of such results is not only to obtain more accurate prediction estimations but also the possibility to visualize the test instances within the generated outcome space.

3.1 Prediction Uncertainty Visualization

For each test instance, the neural network outputs a vector of probability estimates corresponding to the samples generated through Monte Carlo dropout. This creates an opportunity to visualize the variability of individual predictions. With the proposed visualization, we show the correctness and reliability of individual predictions, including false positive results that can be just as informative as correctly predicted ones. The creation of visualizations consists of the following five steps, elaborated below.

1. Projection of the vector of probability estimates into a two dimensional vector space.
2. Point coloring according to the mean probabilities computed by the network.
3. Determining point shapes based on correctness of individual predictions (four possible shapes).
4. Labeling points with respect to individual documents.
5. Kernel density estimation of the projected space—this step attempts to summarize the instance-level samples obtained by the MCD neural network.

As the MCD neural network produces hundreds of probability samples for each target instance, it is not feasible to directly visualize such a multi-dimensional space. To solve this, we leverage the recently introduced UMAP algorithm [19], which projects the input d dimensional data into a s-dimensional (in our case $s = 2$) representation by using computational insights from the manifold theory. The result of this step is a two dimensional matrix, where each of the two dimensions represents a latent dimension into which the input samples were projected, and each row represents a text document.

In the next step, we overlay the obtained representation with other relevant information, obtained during sampling. Individual points (documents) are assigned the mean probabilities of samples, thus representing the reliability of individual predictions. We discretize the $[0, 1]$ probability interval into four bins of equal size for readability purposes. Next, we shape individual points according to the correctness of predictions. We take into account four possible outcomes (TP - true positives, FP - false positives, TN - true negatives, FN - false negatives).

As the obtained two dimensional projection represents an approximation of the initial sample space, we compute the kernel density estimation in this subspace and thereby outline the main neural network's predictions. We use two dimensional Gaussian kernels for this task.

The obtained estimations are plotted alongside individual predictions and represent densities of the neural network's focus, which can be inspected from the point of view of correctness and reliability.

4 Experimental Setting

We first present the data sets used for the evaluation of the proposed approach, followed by the experimental scenario. The results are presented in Sect. 5.

4.1 Hate Speech Data Sets

We use three data sets related to the hate speech.

1 - HatEval data set is taken from the SemEval task "Multilingual detection of hate speech against immigrants and women in Twitter (hatEval)[1]". The competition was organized for two languages, Spanish and English; we only processed the English data set. The data set consists of 100 tweets labeled as 1 (hate speech) or 0 (not hate speech).

2 - YouToxic data set is a manually labeled text toxicity data, originally containing 1000 comments crawled from YouTube videos about the Ferguson unrest in 2014[2]. Apart from the main label describing if the comment is hate

[1] https://competitions.codalab.org/competitions/19935.

[2] https://zenodo.org/record/2586669#.XJiS8ChKi70.

speech, there are several other labels characterizing each comment, e.g., if it is a threat, provocative, racist, sexist, etc. (not used in our study). There are 138 comments labeled as a hate speech and 862 as non-hate speech. We produced a data set of 300 comments using all 138 hate speech comments and randomly sampled 162 non-hate speech comments.

3 - OffensiveTweets data set[3] originates in a study regarding hate speech detection and the problem of offensive language [11]. Our data set consists of 3000 tweets. We took 1430 tweets labeled as hate speech and randomly sampled 1670 tweets from the collection of remaining 23 353 tweets.

Data Preprocessing. Social media text use specific language and contain syntactic and grammar errors. Hence, in order to get correct and clean text data we applied different prepossessing techniques without removing text documents based on the length. The pipeline for cleaning the data was as follows:

- Noise removal: user-names, email address, multiple dots, and hyper-links are considered irrelevant and are removed.
- Common typos are corrected and typical contractions and hash-tags are expanded.
- Stop words are removed and the words are lemmatized.

4.2 Experimental Scenario

We use logistic regression (LR) and Support Vector Machines (SVM) from the scikit-learn library [5] as the baseline classification models. As a baseline RNN, the LSTM network from the Keras library was applied [8]. Both LSTM and MCD LSTM networks consist of an embedding layer, LSTM layer, and a fully connected layer within the Word2Vec and ELMo embeddings. The embedding layer was not used in TF-IDF and Universal Sentence encoding.

To tune the parameters of LR (i.e. *liblinear* and *lbfgs* for the solver functions and the number of component C from 0.01 to 100) and SVM (i.e. the *rbf* for the kernel function, the number of components C from 0.01 to 100 and the gamma γ values from 0.01 to 100), we utilized the random search approach [3] implemented in scikit-learn. In order to obtain best architectures for the LSTM and MCD LSTM models, various number of units, batch size, dropout rates and so on were fine-tuned.

5 Evaluation and Results

We first describe experiments comparing different word representations, followed by sentence embeddings, and finally the visualization of predictive uncertainty.

[3] https://github.com/t-davidson/hate-speech-and-offensive-language.

5.1 Word Embedding

In the first set of experiments, we represented the text with word embeddings (sparse TF-IDF [31] or dense word2vec [21], and ELMo [24]). We utilise the gensim library [27] for word2vec model, the scikit-learn for TFIDF, and the ELMo pretrained model from TensorFlow Hub[4]. We compared different classification models using these word embeddings. The results are presented in Table 1.

The architecture of LSTM and MCD LSTM neural networks contains an embedding layer, LSTM layer, and fully-connected layer (i.e. dense layer) for word2vec and ELMo word embeddings. In LSTM, the recurrent dropout is applied to the units for linear transformation of the recurrent state and the classical dropout is used for the units with the linear transformation of the inputs. The number of units, recurrent dropout, and dropout probabilities for LSTM layer were obtained by fine-tuning (i.e. we used 512, 0.2 and 0.5 for word2vec and TF-IDF, 1024, 0.5, and 0.2 for ELMo in the experiments with MCD LSTM architecture). The search ranges for hyper parameter tuning are described in Table 2.

Table 1. Comparison of classification accuracy (with standard deviation in brackets) for word embeddings, computed using 5-fold cross-validation. All the results are expressed in percentages and the best ones for each data set are in bold.

	HatEval			**YouToxic**			**OffensiveTweets**		
Model	**TF-IDF**	**W2V**	**ELMo**	**TF-IDF**	**W2V**	**ELMo**	**TF-IDF**	**W2V**	**ELMo**
LR	68.0 [2.4]	54.0 [13.6]	62.0 [6.8]	69.3 [3.0]	54.0 [3.0]	**76.6 [6.1]**	**77.2 [1.1]**	68.0 [2.4]	75.6 [1.2]
SVM	63.0 [5.1]	66.0 [3.7]	62.0 [12.9]	70.6 [4.2]	55.0 [3.4]	73.3 [5.5]	77.0 [0.7]	59.6 [1.5]	73.0 [1.9]
LSTM	69.0 [7.3]	67.0 [6.8]	66.0 [12.4]	66.6 [2.3]	59.3 [4.6]	74.3 [2.7]	73.4 [0.8]	75.0 [1.7]	74.7 [1.9]
MCD LSTM	67.0 [10.8]	**69.0 [6.6]**	67.0 [9.8]	66.0 [3.7]	59.3 [3.8]	75.3 [5.5]	71.1 [1.6]	72.0 [1.6]	75.2 [0.9]

Table 2. Hyper-parameters for LSTM and MCD LSTM models

Name	**Parameter type**	**Values**
Optimizers	Categorical	Adam, rmsprop
Batch size	Discrete	4 to 128, step=4
Activation function	Categorical	tanh, relu and linear
Number of epochs	Discrete	10 to 100, step=5
Number of units	Discrete	128, 256, 512, or 1024
Dropout rate	Float	0.1 to 0.8, step=0.05

The classification accuracy for HatEval data set is reported in the Table 1 (left). The difference between logistic regression and the two LSTM models indicates accuracy improvement once the recurrent layers are introduced. On the other hand, as the ELMo embedding already uses the LSTM layer to take into account semantic relationship among the words, no notable difference between logistic regression and LSTM models can be observed using this embedding.

[4] https://tfhub.dev/google/elmo/2.

Results for YouToxic and OffensiveTweets data sets are presented in Table 1 (middle) and (right), respectively. Similarly to the HatEval data set, there is a difference between the logistic regression and the two LSTM models using the word2vec embeddings. For all data sets, the results with ELMo embeddings are similar across the four classifiers.

5.2 Sentence Embedding

In the second set of experiments, we compared different classifiers using sentence embeddings [6] as the representation. Table 3 (left) displays results for HatEval. We can notice improvements in classification accuracy for all classifiers compared to the word embedding representation in Table 1. The best model for this small data set is MCD LSTM. For larger YouToxic and OffensiveTweets data sets, all the models perform comparably. Apart from the prediction accuracy the four models were compared using precision, recall and F1 score [7].

We use the Universal Sentence Encoder module[5] to encode the data. The architecture of LSTM and MCD LSTM contains a LSTM layer and dense layer. With MCD LSTM architecture in the experiments, the number of neurons, recurrent dropout and dropout value for LSTM is 1024, 0.75 and 0.5, respectively. The dense layer has the same number of units as LSTM layer, and the applied dropout rate is 0.5. The hyper-parameters used to tune the LSTM and MCD LSTM models are presented in the Table 2.

Table 3. Comparison of predictive models using sentence embeddings. We present average classification accuracy, precision, recall and F_1 score (and standard deviations), computed using 5-fold cross-validation. All the results are expressed in percentages and the best accuracies are in bold.

	HatEval				YouToxic				OffensiveTweets			
Model	**Accuracy**	**Precision**	**Recall**	**F1**	**Accuracy**	**Precision**	**Recall**	**F1**	**Accuracy**	**Precision**	**Recall**	**F1**
LR	66.0 [12.4]	67.3 [15.3]	65.2 [15.9]	65.2 [13.1]	77.3 [4.1]	74.3 [7.3]	77.3 [3.6]	75.7 [5.3]	80.8 [1.0]	79.6 [1.9]	84.9 [1.2]	82.2 [1.1]
SVM	67.0 [12.1]	68.2 [15.2]	65.0 [15.8]	65.8 [13.3]	77.3 [6.2]	72.6 [8.6]	80.7 [7.4]	76.3 [7.6]	80.7 [1.3]	78.6 [2.0]	86.7 [1.0]	82.4 [1.2]
LSTM	70.0 [8.4]	70.8 [11.0]	63.1 [17.5]	66.2 [14.4]	76.6 [8.6]	73.4 [11.2]	79.2 [8.0]	75.8 [8.6]	80.7 [1.6]	82.8 [2.1]	79.7 [2.3]	81.1 [1.5]
MCD LSTM	**74.0 [10.7]**	73.4 [12.7]	78.4 [13.6]	74.9 [10.0]	**78.7 [5.8]**	74.7 [9.2]	80.9 [6.5]	77.5 [7.4]	**81.0 [1.2]**	81.5 [1.8]	82.5 [2.7]	81.9 [1.3]

5.3 Visualizing Predictive Uncertainty

In Fig. 2 we present a new way of visualizing dependencies among the test tweets. The relations are result of applaing the MCD LSTM network to the HetEval data set. This allows further inspection of the results as well as interpretation of correct and incorrect predictions. To improve comprehensibility of predictions and errors, each point in the visualization is labeled with a unique identifier, making the point tractable to the original document, given in Table 4.

[5] https://tfhub.dev/google/universal-sentence-encoder-large/3.

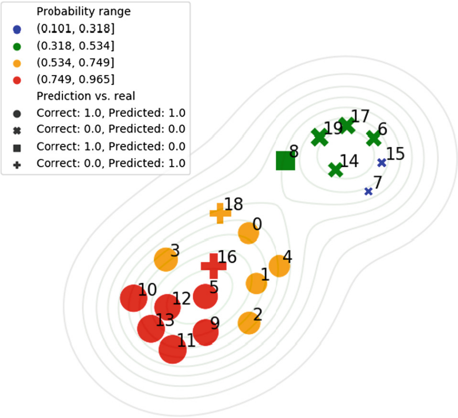


Fig. 2. Visualization of individual predictions along with their probability ranges. The numeric labels besides each point correspond to individual tweets, summarized in Table 4. The size of the points corresponds to the neural network's mean probability for a given prediction. True positives are marked with circles, true negatives with crosses, false positives with squares, and false negatives as pluses.

As Fig. 2 shows, the tweets are grouped into two clusters. According to the kernel density isometric lines, two centers are identified: the tweets assigned lower probability of being hate speech and the tweets with higher probability of being hate speech. Let us focus on the wrongly classified tweets and their positions in the graph (tweets 8, 16 and 18). While for tweets 8 and 18 the classifier wasn't certain and a mistake seems possible according to the plot, the tweet 16 was predicted to be hate speech with high probability. Analyzing the words that form this tweet, we notice that not only that most of them often do appear in the hate speech but also this combination of the words used together is very characteristic for the offensive language.

Table 4. Test documents (one fold) used for the visualization in Fig. 2.

Index (see Figure 2) — Corresponding text
0 — ok if you fu***** said leave block me. but dm me first i am gonna kick your ass. shut the f*** up you
1 — wow mo c*** got hard. want to pull your panties down, push you against these door and let you feel my c*** deep in your wet p****
2 — no i said hope. i hope you women learn your place! sitdowninthekitchen
3 — so this b**** blocked me but check her pinned tweet out, she is accusing men of rape on tumblr with . . .
4 — this is not rape, or anything like it. this a crude pass. this is not 'rape culture.' it is w**** culture.
5 — rape is bad but alota h*** be lying bout that s*** women are crazy dawg & will f*** ya life up be careful
6 — lol. they did not applaud a rape threat .how long are you going to ride that s***?
7 — you had a great chance to show everyone that youre not a c*** on tv tonight and you f***** it up.
8 — lmao!!! stuff i say. some stuff be so damn funny.especially when it comes to illegal immigrants who are full fledged criminals coming to america . . .
9 — the german government pays for 3 week vacation for refugees to go home -
10 — burundian refugees should go home!
11 — no benefits to any immigrants !! come here &amp; make it on your own, or go back home !!
12 — illegal aliens waiting for first class msc container-shippping in antwerpen stop immigration
13 — sick barstewards! this is what happens when we put up the refugees welcome signs! they not only rape our wives or girlfriends . . .
14 — i am not forcing anyone. i am just putting out my view. people believe it or not is their prerogative.even if a single person agree with me . . .
15 — tajikistani charged with beating new york university professor to death another animal./
16 — child prostitution: feds break up mexico-to-queens sex trafficking ring via . . .
17 — home office guilty of a serious breach of the duty of candour and cooperation regarding children entitled to enter uk. where did these children go? . . .
18 — p.s why do you not pay unemployed people who do endless hours of voluntary work they do that to give something to the community
19 — seriously, amy and cindy are bffs, i know that for sure. hmm, mmm.

Our short demonstration shows the utility of the proposed visualization which can identify different types of errors and helps to explain weaknesses in the classifier or wrongly labeled data.

6 Conclusions

We present the first successful approach to assessment of prediction uncertainty in hate speech classification. Our approach uses LSTM model with Monte Carlo dropout and shows performance comparable to the best competing approaches using word embeddings and superior performance using sentence embeddings. We demonstrate that reliability of predictions and errors of the models can be comprehensively visualized. Further, our study shows that pretrained sentence embeddings outperform even state-of-the-art contextual word embeddings and can be recommended as a suitable representation for this task. The full Python code is publicly available[6].

As persons spreading hate speech might be banned, penalized, or monitored not to put their threats into actions, prediction uncertainty is an important component of decision making and can help humans observers avoid false positives and false negatives. Visualization of prediction uncertainty can provide better understanding of the textual context within which the hate speech appear. Plotting the tweets that are incorrectly classified and inspecting them can identify the words that trigger wrong classifications.

Prediction uncertainty estimation is rarely implemented for text classification and other NLP tasks, hence our future work will go in this direction. A recent emergence of cross-lingual embeddings possibly opens new opportunities to share data sets and models between languages. As evaluation in rare languages is difficult, the assessment of predictive reliability for such problems might be an auxiliary evaluation approach. In this context, we also plan to investigate convolutional neural networks with probabilistic interpretation.

Acknowledgments. The work was partially supported by the Slovenian Research Agency (ARRS) core research programme P6-0411. This project has also received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825153 (EMBEDDIA).

References

1. Baldi, P., Sadowski, P.J.: Understanding dropout. In: Advances in Neural Information Processing Systems, pp. 2814–2822 (2013)
2. Berger, W., Piringer, H., Filzmoser, P., Gröller, E.: Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. In: Computer Graphics Forum, pp. 911–920 (2011)
3. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**, 281–305 (2012)

[6] https://github.com/KristianMiok/Hate-Speech-Prediction-Uncertainty.

4. Bleich, E.: The rise of hate speech and hate crime laws in liberal democracies. J. Ethnic Migr. Stud. **37**(6), 917–934 (2011)
5. Buitinck, L., et al.: API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning, pp. 108–122 (2013)
6. Cer, D., et al.: Universal sentence encoder. arXiv preprint arXiv:1803.11175 (2018)
7. Chinchor, N.: Muc-4 evaluation metrics. In: Proceedings of the Fourth Message Understanding Conference, p. 22–29 (1992)
8. Chollet, F., et al.: Keras (2015). https://keras.io
9. Corazza, M., et al.: Comparing different supervised approaches to hate speech detection. In: EVALITA 2018 (2018)
10. Cox, J., Lindell, M.: Visualizing uncertainty in predicted hurricane tracks. Int. J. Uncertain. Quantif. **3**(2), 143–156 (2013)
11. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated hate speech detection and the problem of offensive language. In: Eleventh International AAAI Conference on Web and Social Media (2017)
12. Del Vigna12, F., Cimino23, A., Dell'Orletta, F., Petrocchi, M., Tesconi, M.: Hate me, hate me not: Hate speech detection on facebook (2017)
13. Fortunato, M., Blundell, C., Vinyals, O.: Bayesian recurrent neural networks. arXiv preprint arXiv:1704.02798 (2017)
14. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In: International Conference on Machine Learning, pp. 1050–1059 (2016)
15. Gal, Y., Ghahramani, Z.: A theoretically grounded application of dropout in recurrent neural networks. In: Advances in Neural Information Processing Systems, pp. 1019–1027 (2016)
16. Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., Blei, D.M.: Automatic differentiation variational inference. J. Mach. Learn.Res. **18**(1), 430–474 (2017)
17. Liu, L., et al.: Uncertainty visualization by representative sampling from prediction ensembles. IEEE Trans. Vis. Comput. Graph. **23**(9), 2165–2178 (2016)
18. Liu, L., Padilla, L., Creem-Regehr, S.H., House, D.H.: Visualizing uncertain tropical cyclone predictions using representative samples from ensembles of forecast tracks. IEEE Trans. Vis. Comput. Graph. **25**(1), 882–891 (2019)
19. McInnes, L., Healy, J., Saul, N., Grossberger, L.: UMAP: Uniform manifold approximation and projection. J. Open Source Softw. **3**(29), 861 (2018)
20. Mehdad, Y., Tetreault, J.: Do characters abuse more than words? In: Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 299–303 (2016)
21. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
22. Miok, K.: Estimation of prediction intervals in neural network-based regression models. In: 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), pp. 463–468, September 2018
23. Myshkov, P., Julier, S.: Posterior distribution analysis for Bayesian inference in neural networks. In: Workshop on Bayesian Deep Learning, NIPS (2016)
24. Peters, M.E., et al.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
25. Pham, V., Bluche, T., Kermorvant, C., Louradour, J.: Dropout improves recurrent neural networks for handwriting recognition. In: 2014 14th International Conference on Frontiers in Handwriting Recognition, pp. 285–290. IEEE (2014)

26. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in large margin classifiers, pp. 61–74. MIT Press (1999)
27. Rehurek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pp. 45–50. ELRA, Valletta, Malta, May 2010
28. Rother, K., Allee, M., Rettberg, A.: Ulmfit at germeval-2018: a deep neural language model for the classification of hate speech in German tweets. In: 14th Conference on Natural Language Processing KONVENS 2018, p. 113 (2018)
29. Ruginski, I.T., et al.: Non-expert interpretations of hurricane forecast uncertainty visualizations. Spat. Cogn. Comput. **16**(2), 154–172 (2016)
30. Schmidt, A., Wiegand, M.: A survey on hate speech detection using natural language processing. In: Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, pp. 1–10 (2017)
31. Sparck Jones, K.: A statistical interpretation of term specificity and its application in retrieval. J. Doc. **28**(1), 11–21 (1972)
32. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
33. Waldron, J.: The Harm in Hate Speech. Harvard University Press, Cambridge (2012)
34. Wang, S., Manning, C.: Fast dropout training. In: International Conference on Machine Learning, pp. 118–126 (2013)
35. Warner, W., Hirschberg, J.: Detecting hate speech on the world wide web. In: Proceedings of the Second Workshop on Language in Social Media, pp. 19–26. Association for Computational Linguistics (2012)
36. Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent neural network regularization. arXiv preprint arXiv:1409.2329 (2014)
37. Zhu, L., Laptev, N.: Deep and confident prediction for time series at uber. In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 103–110. IEEE (2017)

Appendix C: Automating news comment moderation with limited resources: Benchmarking in Croatian and Estonian

Ravi Shekhar, Marko Pranjić, Senja Pollak, Andraž Pelicon, Matthew Purver

Automating News Comment Moderation with Limited Resources: Benchmarking in Croatian and Estonian

This article describes initial work into the automatic classification of user-generated content in news media to support human moderators. We work with real-world data — comments posted by readers under online news articles — in two less-resourced European languages, Croatian and Estonian. We describe our dataset, and experiments into automatic classification using a range of models. Performance obtained is reasonable but not as good as might be expected given similar work in offensive language classification in other languages; we then investigate possible reasons in terms of the variability and reliability of the data and its annotation.

1. Introduction

This article describes initial work on the EMBEDDIA project into the automatic classification of user-generated content (UGC) in news media: reader comments posted under news articles. The EMBEDDIA project focuses on the use of cross-lingual techniques to transfer language technology resources to less-resourced languages (as well as English and Russian, the project focuses on Slovene, Croatian, Estonian, Lithuanian, Latvian, Finnish, and Swedish), and the application of these to real-world problems in the news media industry. One such problem is the need for news publishers to allow readers to post comments under articles online, in order to promote engagement with the content, but prevent content being published that would be offensive to other readers, dangerous or in some way compromise the legal position of the publisher. Most publishers currently use manual methods to do this: a team of moderators will monitor comments and block them when required. However, high volumes of comments can often make this impractical. The use of automatic natural language processing methods to detect comments that should be blocked, or referred to human moderators, can speed up the process many times (Pavlopoulos et al., 2017a); and many successful approaches to automated hate and offensive speech detection and categorisation exist (see e.g. MacAvaney et al., 2019; Schmidt & Wiegand, 2017), with datasets and shared tasks made available for several major EU languages (see e.g. Zampieri et al., 2019; V. Basile et al., 2019). However, such resources are generally only available for a few languages (e.g., English, German), leaving a gap for less-resourced languages. For Estonian and Croatian, languages of interest here, the number of studies is very limited (Ljubešić et al., 2018).

In this work, we describe new data collection efforts in two less-resourced European languages (Croatian, Estonian), and our experiments into automated classification. We explain the existing moderation scheme used by humans in news editorial houses, and examine to what extent it overlaps with the concept of offensive language as usually defined; describe a range of suitable classifier architectures for automatic detection of problematic comments; and give results showing that although reasonable performance can be achieved on these languages given suitable methods, it does not reach the levels that might be expected given other related work in languages in which more resources are available. We then examine the robustness of both the classifiers and the moderation scheme itself, and find that performance is limited not only by the nature of interactive language and its dependence on context, but by the need to rely on labels gathered under real-world constraints. We conclude that a transfer learning approach is the most promising future direction, providing the opportunity to incorporate information from more, better-curated datasets available in other languages, but that this will require cross-lingual techniques beyond the current state of the art.

2. Data and Task

The task of interest here, broadly defined, is to develop an automatic classifier to automate (or partially automate) the manual process of moderation: deciding which reader comments should be blocked, according to the policy of a particular newspaper.

2.1. Dataset

For this work, we have collected a large new dataset of online reader comments, from a range of news media sources in two less-resourced European languages, as covered by our project partners. Our dataset consists of over 60 million comments from the articles published online by three major news outlets:

- **24sata (`www.24sata.hr`):** The largest-circulation daily newspaper in Croatia, reaching on average 2 million readers daily.[1] Language: Croatian. Size: 21.5M comments.
- **Večernji List (`www.vecernji.hr`):** The third-largest daily newspaper in Croatia. Language: Croatian. Size: 9.6M comments.
- **Eesti Ekspress (`www.ekspress.ee`):** The largest weekly newspaper in Estonia, with a circulation of over 20,000. Languages: Estonian, Russian (articles are written in Estonian, but comments are often also in Russian). Size: 31.5M comments.

2.2. Annotation

In each case, the comments are annotated with metadata including link to the relevant article, ID of the comment author (anonymised) and timestamp; importantly for the purposes of this work, comments are also labelled if they are blocked by human moderators. Details of the moderation policy, and therefore the nature of the labelling, vary with news source, but comments may be blocked for a wide range of reasons. For 24sata, the annotation reflects a moderation policy based on 8 different categories, shown in Table 1; comments should be blocked if they breach any one of these categories, although the implications for the comment author vary with the severity of the category. Less serious offences (labelled 'minor' in Table 1) lead to a minor warning: a user may receive up to two minor warnings, but the third one leads to a temporary one-day ban from the site.

[1] https://showcase.24sata.hr/2019_hosted_creatives/medijske-navike-hr-2019.pdf

More serious offences lead to major warnings, of which a user may only receive one – the second one leads to a five-day ban. After a ban, the number of warnings of that type are reset to zero, but breaking the rules multiple times can, at the discretion of the moderators, lead to a permanent ban.

Rule ID	Description	Definition	Severity
1	Disallowed content	Advertising, content unrelated to the topic, spam, copyright infringement, citation of abusive comments or any other comments that are not allowed on the portal	Minor
2	Threats	Direct threats to other users, journalists, admins or subjects of articles, which may also result in criminal prosecution	Major
3	Hate speech	Verbal abuse, derogation and verbal attack based on national, racial, sexual or religious affiliation, hate speech and incitement	Major
4	Obscenity	Collecting and publishing personal information, uploading, distributing or publishing pornographic, obscene, immoral or illegal content and using a vulgar or offensive nickname that contains the name and surname of others	Major
5	Deception & trolling	Publishing false information for the purpose of deception or slander, and “trolling” - deliberately provoking other commentators	Minor
6	Vulgarity	Use of bad language, unless they are used as a stylistic expression, or are not addressed directly to someone	Minor
7	Language	Writing in other language besides Croatian, in other scripts besides Latin, or writing with all caps	Minor
8	Abuse	Verbally abusing of other users and their comments, article authors, and direct or indirect article subjects, calling the admins out or arguing with them in any way	Minor

Table 1: Annotation schema for blocked comments, 24sata.

As Table 1 shows, the categories cover a broad range of grounds for moderation, and many categories potentially overlap. They include a range of categories in the broad area of offensive language, many of which might overlap: threats to others (rule 2); hate speech based on national, racial, sexual or religious affiliation (3); obscene or immoral content (4); bad language (6); and verbal abuse (8). However, they also include a range of other reasons: illegal content (rule 1); comments not allowed by the portal’s rules (1); advertising (1); off-topic posts (1); copyright infringement (1); false information (5); use of language other than Croatian (7).

Furthermore, as Table 2 shows, these categories also vary between publishers: the categories for Večernji List (hereafter VL) have many similarities with those for 24sata, but it is not possible to map directly between them. Categories such as hate speech and threats seem to correspond directly (rules 3 and 2 for 24sata, rules 1 and 2 for VL); but others are combined in different ways

Rule ID	Corresponding 24sata rule ID(s)	Definition	Severity
1	3	Hate speech on a national, religious, sexual or any other basis	Major
2	2	Threats to other users, administrators, journalists or subjects of articles	Major
3	6, part 4, part 8	Insulting other users or use of bad language.	Minor
4	part 4	Publishing personal data	Minor
5	part 1, part 7	Chat, off-topic, writing in all caps, posting links	Minor
6	part 7	Writing in a script other than a Latin script	Minor
7	part 8	Challenging the administrators or arguing with then in any way	Minor
8	part 5	Posting false information	Minor
9	n/a	Using multiple user accounts	Permanent ban

Table 2: Annotation schema for blocked comments, Večernji List, together with corresponding Rule IDs from the 24sata schema (Table 1).

(e.g. 24sata's rule 5 covers posting false information, which maps to VL's rule 8, but also covers trolling and povocation which does not seem to be explicitly covered in VL's policy; VL's rule 3 covers insults and bad language, aspects of which are covered by parts of 24sata's rules 4, 6 and 8). Ekspress, on the other hand, do not record explicit categories of policy violation, so no such detailed annotation is available.

Three distinct problems therefore arise. First, distinguishing between the categories — rather than just detecting the general category of requiring moderation — is an important task in order to record how the policy was applied when blocking a comment or banning a user, where such a policy exists. Second, the overall category of blocked comments is likely to cover a very heterogeneous sample of language, as it results from a diverse range of phenomena. Third, as the categories are not *a priori* fixed, and can be conceptually divided up in different ways, this heterogeneity is likely to extend even to the individual classes.

Problematic comments are fairly common: for the 24sata subset, articles receive around 45 comments on average, and those that receive problematic comments receive around 5.5 of them. However, the data is highly unbalanced — only around 5-6% of comments require blocking — bringing an added complication to the classification task.

3. Related Work and Resources

In this section, we investigate what resources might be available which can help; in particular, what datasets might be available to provide training data for suitable classifiers.

3.1. Comment Filtering

Previous work in news comment filtering is limited. Pavlopoulos et al. (2017a) address the problem using data from a Greek newspaper, Gazzetta. They use a dataset of 1.6M comments with labels derived from the newspaper's human moderators and journalists; they test a range of neural network-based classifiers and achieve encouraging performance with AUC scores (area under the ROC curve) of 0.75-0.85 depending on the data subset. However, being in a different language (Greek) their data is not directly usable as a training set for our task. In addition, their moderation labels are binary, representing a "block or not" decision, rather than giving any further information about the reasons behind a decision. They are therefore not suited to investigating the moderation policy labels of interest here; and more fundamentally, it is unclear whether the decisions of Gazzetta's moderators are based on similar aims or policies as the decisions we must try to simulate for 24sata or Ekspress's moderators. Pavlopoulos et al. (2017a) asked additional annotators to classify comments according to a more detailed taxonomy (*"We also asked the annotators to classify each snippet into one of the following categories: calumniation (e.g., false accusations), discrimination (e.g., racism), disrespect (e.g., looking down at a profession), hooliganism (e.g., calling for violence), insult (e.g., making fun of appearance), irony, swearing, threat, other."*) but this was done as a post-hoc exercise and only for a small portion of the test set. It was not used in classification experiments, but only for separate analysis purposes.

Other work with reader comments on news (see Table 3) exists but does not attempt to learn from or reproduce moderation decisions directly in the same way. Kolhatkar et al. (2019) and Napoles et al. (2017) investigate constructivity in comments, and provide datasets which distinguish between constructive and non-constructive comments; these datasets are related to our task, though, as they also include information about toxicity and related categories such as insults and off-topic posting. Barker et al. (2016) investigate quality of comments and their use in summarisation. Wulczyn et al. (2017) investigate a related problem of detection of personal attacks and toxicity in user comments on Wikipedia articles, rather than news; and Zhang et al. (2018) also investigate Wikipedia comments from the point of view of detecting which conversations become toxic. None of these directly solve our problem, although they could in theory provide useful information; however, all are limited to English data.

3.2. Resources for Related Tasks

A variety of related tasks have been studied in data other than user-generated comments on articles. Given the moderation policy details in Section 2 above, the existence of suitable datasets for training classifiers for various categories of offensive language, advertising/spam, and trolling behaviour would be of interest. While none of these categories corresponds directly to the overall category of comments that must be blocked, each one covers a phenomenon that requires blocking.

3.2.1. Offensive Language Detection

Recent years have seen a large amount of research on detection of offensive language of various kinds. Many public datasets have been created and distributed, many shared tasks have been run, and many classification systems developed and tested (see Table 4). The exact definition of the

Corpus	Location	Domain	Language	Size	Type of annotation
Gazzetta	(Pavlopoulos et al., 2017a)	News	gr	1.6M	Moderation
SFU SOCC	(Kolhatkar et al., 2019)	News	en	663k	Constructiveness, toxicity
YNACC	(Napoles et al., 2017)	News	en	522k	Constructiveness, insults, off-topic
SENSEI	(Barker et al., 2016)	News	en	2k	Quality, tone, summaries
DETOX	(Wulczyn et al., 2017)	Wiki	en	115k	Personal attacks, aggression, toxicity
Zhang et al., 2018	(Zhang et al., 2018)	Wiki	en	7k	Personal attacks

Table 3: Existing datasets for filtering user-generated comments on articles. Size is given in number of comments.

Corpus	Location	Domain	Language	Type of annotation
FRENK	(Ljubešić et al., 2019)	Facebook	en,sl	Socially unacceptable language
HASOC	hasoc2019.github.io	Twitter/Facebook	de, en, hi	Hate speech, target
HatEval 2019	(V. Basile et al., 2019)	Twitter	en, es	Hate speech, target, aggression
OLID (OffensEval)	(Zampieri et al., 2019)	Twitter	en	Hate speech, target, threats
GermEval	(Wiegand et al., 2018)	Twitter	de	Abuse, profanity, insults
IBEREVAL	(Anzovino et al., 2018)	Twitter	en,es	Misogynous
MEX-A3T	(Álvarez-Carmona et al., 2018)	Twitter	es-mx	Aggressive
Liu et al 2018	(Liu et al., 2018)	Instagram	en	Hostile
Waseem & Hovy 2016	(Waseem & Hovy, 2016)	Twitter	en	Hate speech, with subcategory
Stormfront	(de Gibert et al., 2018)	Online forum	en	White supremacy

Table 4: Existing datasets: abuse, hate speech and offensive language. "Target" refers to annotation of the group or individual towards which hate speech is directed.

categories annotated in these tasks varies, however (see Schmidt & Wiegand, 2017, for a survey), and may include one or all of:

- Threats: hostile speech intended to threaten the addressee with violence or other negative effects;
- Abuse: personal insults directed at others, including 'flaming' or cyberbullying;
- Hate speech: personal attacks on the basis of religion, race, sex, sexuality etc.;
- Offensive content: the use of language which is in itself considered rude, vulgar or profane (including pornographic), even if not targeted at someone in particular.

These terms are often used interchangeably, with some (particularly *hate speech*) often used to cover multiple categories. Exact definitions of the individual categories also vary with task and dataset, so we do not attempt an exhaustive exposition here. As an illustrative example, Waseem & Hovy (2016) define their *hate speech* category for Twitter as a message that:

1. *uses a sexist or racial slur;*
2. *attacks a minority;*
3. *seeks to silence a minority;*
4. *criticizes a minority (without a well founded argument);*
5. *promotes, but does not directly use, hatespeech or violent crime;*
6. *criticizes a minority and uses a straw man argument;*
7. *blatantly misrepresents truth or seeks to distort views on a minority with unfounded claims;*
8. *shows support of problematic hash tags. E.g.“#BanIslam”, “#whoriental”, “#whitegenocide”;*
9. *negatively stereotypes a minority;*
10. *defends xenophobia or sexism;*
11. *contains a screen name that is offensive, as per the previous criteria, the tweet is ambiguous (at best), and the tweet is on a topic that satisfies any of the above criteria.*

On the other hand, Ljubešić et al. (2019) use a more restrictive set of definitions via a decision tree to separate out different kinds of *socially unacceptable discourse (SUD)* on Facebook into different categories:

Is this SUD aimed at someone’s background?
- YES: Are there elements of violence?
  - YES: background, violence
  - NO: background, offensive speech
- NO: Is this SUD aimed towards individuals or other groups?
  - YES: Are there elements of violence?
    - YES: other, threat
    - NO: other, offensive speech
  - NO: Is the speech unacceptable?
    - YES: inappropriate speech
    - NO: acceptable speech

In all these variants, the task is usually defined as a classification task — detecting whether a given text should be classified as hate speech (or abuse, offensive language etc.) or not — although this may be set up as a binary or a multi-class classification problem depending on the definitions used. Many datasets are available for this broad category of tasks, with a number of public shared tasks having been run over the last few years.[2] The exact categories annotated vary, as do the domain and language of text annotated; we give an indication of each in Table 4.

Most datasets are based on social media (mainly Twitter) posts. Performance varies widely with dataset and domain. OffensEval 2019 reports maximum F1 score 0.829 on the offense classification task; for the white supremacy forum comments (de Gibert et al., 2018) classification accuracy is 0.78.

3.2.2. Spam detection

Another important task for UGC filtering in many domains, corresponding to one of the categories in the 24sata moderation policy in Section 2, is the detection of *spam*: comments which are off-topic, intended not to contribute to an ongoing conversation or relate to a given topic but rather to advertise, and/or to entice readers into clicking on a link either to generate revenue or for more nefarious purposes (e.g. 'phishing', attempting to gain access to personal information). This task is highly relevant for news media companies in order to prevent comments sections being taken over by irrelevant, offputting or dangerous content.

The task is a variant of the familiar spam detection problem for email (see Caruana & Li, 2012, for a survey), but UGC and online comments have their own distinctive characteristics – see for example (Kantchelian et al., 2012) for application to comments in the blog domain, (Aiyar & Shetty, 2018) in the Youtube domain, and (Wu et al., 2018) for a survey of work in the Twitter domain.

Corpus	Location	Size	Language	Domain
NSC Twitter Spam	(Chen et al., 2015)	6 million tweets	en	Twitter
Youtube Spam Collection	(Alberto et al., 2015)	1956 comments	en	Youtube
MPI-SWS	(Ghosh et al., 2012)	41,352 accounts	n/a	Twitter

Table 5: Existing datasets: spam.

Corpus	Location	Size	Language	Domain
FiveThirtyEight	(Linvill & Warren, 2018)	2,973,371 tweets	en	Twitter
Dataturks	(Narayanan, 2018)	20,000 tweets	en	Social media
Mojica 2017	(Mojica de la Vega & Ng, 2018)	5,868 conversations	en	Reddit

Table 6: Existing datasets: trolling and incitement.

Table 5 shows a sample of the most relevant datasets here. Alberto et al. (2015) provide a dataset of comments on Youtube videos classified as spam or not. Several datasets are available for short text messages in social media, see e.g. (Chen et al., 2015)'s large collection of 6 million spam tweets, and the MPI collection of Twitter accounts detected as spam accounts. Again, this task is usually defined as a binary classification task. Performance varies widely with dataset and

[2] A helpful catalogue of relevant datasets is also available online at `http://hatespeechdata.com/`.

domain. Wu et al. (2018) report accuracies of up to 94.5% on account classification and 88-91% accuracy on individual texts.

3.2.3. Trolling and incitement

Another basis for moderation in the policy of Section 2 is the presence of *trolls* and *bots*: users who may be automated or semi-automated rather than human, and which behave in a disruptive and/or deceptive manner in order to influence discussion, spread propaganda and manipulate opinion or to incite extreme views and disrupt discussion (see e.g. Kim et al., 2019). The effects of such agents in social media and news article comments can be strong, with evidence that they have affected public opinion and outcomes of elections (Badawy et al., 2018). There is a connection with the *fake news* phenomenon, with many trolling accounts being used to spread false rumours and link to fake news.

In this case, although this can be approached in a similar classification manner to the tasks above, labelling texts as coming from trolls, the problem is more often seen as one of classifying user accounts rather than their individual text outputs. Methods used therefore often depend as much on the social network properties of user accounts as on the language they generate. Again, some datasets exist; see Table 5. FiveThirtyEight distribute a dataset of nearly 3 million tweets sent from Twitter accounts *"connected to the Internet Research Agency, a Russian "troll factory" and a defendant in an indictment filed by the Justice Department in February 2018"* between February 2012 and May 2018. Narayanan (2018) then provides a smaller dataset from the same source, but annotated in more detail for level of aggression. Mojica (2017); Mojica de la Vega & Ng (2018) collected a similar dataset of comments on Reddit.

In our domain of UGC comments under news articles, Mihaylov & Nakov (2016) collected a dataset from over 2 years of articles (Jan 2013-April 2015) on the Bulgarian news site Dnevnik (`dnevnik.bg`), totalling 1,930,818 comments by 14,598 users on 34,514 articles. Troll comments were identified by a combination of observing other users' reactions, and checking identities in leaked documents; however, the dataset is not currently available publicly.

Mihaylov & Nakov (2016) achieve around 81% accuracy and F-score on the classification task, on a balanced dataset of news comments, using simple baseline linear classifiers. Mojica (2017) achieves c.90% accuracy on his dataset for the trolling detection task, using a more complex conditional random field classifier.

3.3. The Problem of Monolinguality

As the discussion above shows, datasets are available. However, very few are in the exact domain of automatic moderation: the Gazzetta dataset of (Pavlopoulos et al., 2017b) is the only example from news, with the Wikipedia dataset of (Wulczyn et al., 2017) being quite closely related. More critically, none are available in the languages required here (Croatian, Estonian); the closest are the Facebook dataset of socially unacceptable discourse in Slovenian of Ljubešić et al. (2019), and the Bulgarian news comment trolling data of Mihaylov & Nakov (2016), but neither are publicly available and neither are in the exact domain required.

This problem is a widespread one in NLP: a large majority of research and available datasets is monolingual and in English, and datasets for specific less-resourced languages like Croatian and Estonian are hard to find. Some multi-lingual work exists: Ousidhoum et al. (2019) present a multilingual hate speech study on English, French and Arabic tweets, and A. Basile & Rubagotti (2018) conduct cross-lingual experiments between Italian and English; again, this does not cover our languages or domain.

We also note the existence of Hatebase,[3] a highly multilingual collection of crowdsourced social media posts; however, as its annotation is based only on submission by the public, and it contains no comparable non-abuse language, it is not currently suitable as training or evaluation data for a classifier of the kind needed here.

We therefore conclude that for our present purposes, training on the specific data we have, in the correct language and reflecting the moderation policy of the correct newspaper, is the only practical option. The next section outlines our experiments using this approach.

4. Experiments

Our approach is therefore to treat the task as a classification problem, and use the real-world moderator decisions, recorded in the newspaper databases, as our training and test labels.

4.1. Classification Models

We formulate the problem as a text classification task. The basic task is a binary choice: given a comment, a system has to predict whether it should be *blocked* or *non-blocked*. We can also consider a multi-class task: given a comment, to predict which rule (Table 1 or Table 2) is being violated. We compared four different models, each using a standard method for text classification.

Naïve Bayes As a baseline, we use a Naïve Bayes (NB) classifier. NB is a simple probabilistic generative model which makes the approximation that words are independent of one another: the probability of a text belonging to a particular class can therefore be approximated as the product of the probabilities of the individual constituent words being associated with that class, and those can be calculated directly from frequencies in the training set. While clearly an oversimplification, this approach can provide good results in many text classification tasks, including spam detection (see e.g. Jurafsky & Martin, 2009). It also provides an easily interpretable model: a conditional probability table relating each word to each class.

LSTM In this model, the comment is encoded using a Long Short-Term Memory (LSTM) recurrent neural network (Hochreiter & Schmidhuber, 2015): LSTMs are able to encode not only word sequence but capture dependencies between non-adjacent words. The last hidden state of the LSTM is taken as the representation of the comment, and on top of that, a multi-layer perceptron (MLP) is used to produce the classification decision. Word embedding vectors are randomly initialised, and the whole architecture is trained end-to-end.

[3] `http://hatebase.org/`

LASER In this model, the comment is represented using Language-Agnostic SEntence Representation (LASER, Artetxe & Schwenk, 2019). LASER produces representations for sentence-length texts, obtained using a five-layer bidirectional LSTM (BiLSTM) encoder with a shared byte-pair encoded (BPE) dictionary for 92 languages. The last states of the LSTM are used to produce a sentence vector by max-pooling, and the model is trained using an encoder-decoder approach, in which the sentence representations are used to generate parallel sentences in another language. This approach gives sentence vectors which capture many aspects of sentence meaning and can be used in many tasks; here, we use a MLP on top of the sentence representations, and train it on our classification task. Only the MLP is trained; the weights of the LASER encoder are kept frozen using the pre-trained models available.[4]

mBERT In our final model, the comments are represented using Bidirectional Encoder Representations from Transformers (BERT, Devlin et al., 2018). BERT is a deep contextual representation based on a series of layers of Transformer cells (Vaswani et al., 2017), and trained using a variant of a language model objective. As with LASER above, we then pass the comment representation to a MLP for classification. The BERT model weights are initialized using the multilingual pre-trained model (mBERT, trained on 104 languages by sharing embeddings across languages), and fine-tuned end-to-end along with the MLP.[5]

Training Note the difference in the training strategy for our LSTM, LASER, and mBERT models. In the case of LSTM, the whole architecture is initialized randomly and trained end-to-end: we use no pre-trained embeddings, and train only on the data available here. In the case of LASER, only the classification MLP weights are trained, while the LASER model sentence (comment) representation weights are kept fixed at the values in the pre-trained model. For mBERT, the comment representation weights are initialized using the pre-trained model, and the MLP weights initialized randomly, and the whole model is then fine-tuned end-to-end. All the neural models are trained using the Adam optimizer (Kingma & Ba, 2014) with cross-entropy loss.

4.2. Experiment 1: Binary Classification

4.2.1. Data Selection

As Figure 1 shows, the rate of commenting on articles, and the rate at which moderators block comments, vary over time. (Detailed frequency counts are given in Appendix A, Section A.1). For Ekspress, the rate of commenting rises steadily over time; for 24sata, it rises to a peak in 2015/2016 and then reduces slightly. For VL, the commenting rate seems more stable. (Note that the data was collected part-way through the year 2019, so data for that year is not for a complete year period). Particularly of note, though, is that the rate at which moderators block comments rises over time for all newspapers; the effect is particularly marked for VL from 2013 onwards, and for 24sata from 2016 onwards. Note that the rates for VL before 2013, and 24sata before 2016, are not zero, but very low; see Appendix A for details. This effect is not merely one of

[4] Pre-trained model available from `https://github.com/facebookresearch/LASER`.
[5] Pre-trained model available from `https://github.com/google-research/bert`.

comment volume: higher commenting rates do not correspond to higher blocking rates (Figure 1), as might be hypothesized if, say, a rise in commenting rates were caused by a sudden influx of troll accounts or an increase in contentious topics. Instead, the most likely cause is a change in moderation policy: over recent years, more attention has been given by newspapers to moderation, in terms of both overall importance and strictness of adherence to policy. Note also that blocking rates are relatively low in general: even the peak rate for VL is only just over 15% of comments, for Ekspress 12.5%, and for 24sata only 7.8%: this gives an unbalanced dataset which must be accounted for in training and testing.

Given the sharp change over time, it seems very likely that data from more recent years will be more consistent, and will be more reflective of current moderation policy: earlier years are likely to contain large numbers of false negatives (comments that were not moderated at the time, due to either lack of resources or difference in policy, but would be blocked now). In order to have the cleanest and most relevant data possible, we therefore first selected 2019 data for training, validation, and testing purposes. Since most comments are non-blocked comments, to have a balanced dataset for experiment purposes, we first selected only those articles which have at least one blocked comment. We then divided those articles into training (80%), validation (10%) and test (10%) partitions. Finally, we randomly selected an equal number of blocked and unblocked comments per article in each set. Table 7 shows the resulting data distribution for all three newspapers.

	24sata			Večernji List			Ekspress		
	Train	Val	Test	Train	Val	Test	Train	Val	Test
# Articles	9196	1148	1154	6521	813	821	7490	934	942
# Comments	99246	12364	12472	85916	10490	10855	145154	19310	20312

Table 7: Partitioned dataset distribution, 24sata, Večernji List and Ekspress.

4.2.2. Results

Table 8 shows the results for each classifier model. As our training and test sets have an evenly weighted number of positive (blocked) and negative (non-blocked) examples, we give performance as standard percentage accuracy, and to get an insight into the relative performance we give this not only overall but over the positive and negative portions of the test set individually. 'Blocked' accuracy is therefore equivalent to recall for the positive (blocked) class; 'Non-blocked' accuracy is recall for the negative (non-blocked) class. Standard summary measures such as weighted average F-score are not very helpful in this setting, as they can be so strongly dominated by the majority (non-blocked) class, and accuracy on the two classes has different implications for news publishers; we therefore examine per-class metrics (although see Section 4.4 for results in terms of macro-averaged F-score on the final dataset).

For all three newspapers, the mBERT model gives best performance. Surprisingly, the NB model gives relatively strong performance, with neither the LSTM nor LASER models providing much of an improvement; in fact, for Ekspress they perform worse than NB. Accuracy is higher for 24sata than for Ekspress and VL, but in all cases the absolute level of accuracy is lower than might

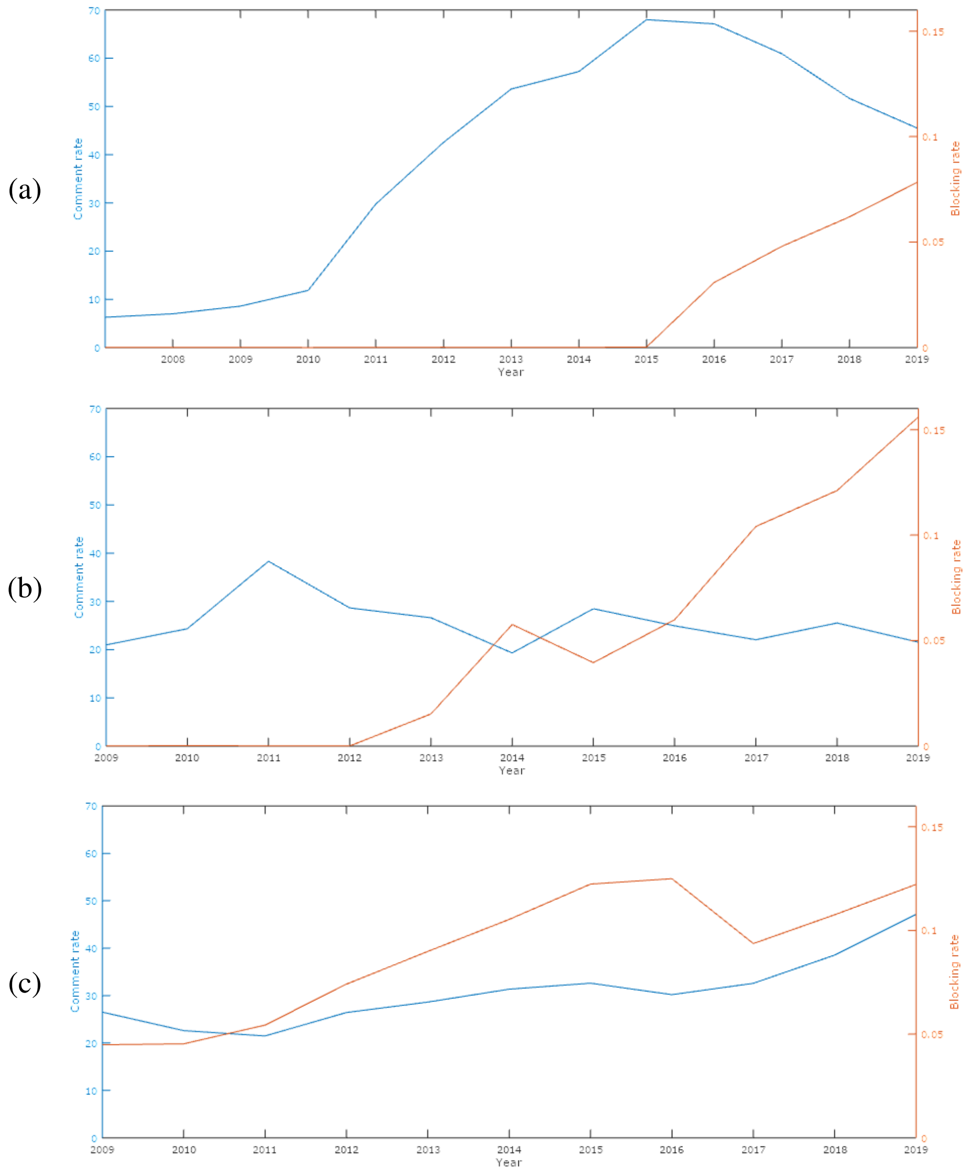


Figure 1: Comment rate $N_{\text{comments}}/N_{\text{articles}}$ in blue, and blocking rate $N_{\text{blocked}}/N_{\text{comments}}$ in red, over time, for (a) 24sata, (b) Večernji List, (c) Ekspress.

be expected given comparable experiments with offensive language detection in other research (Section 3). Accuracy on blocked content is lower than the accuracy of recognition of non-blocked content, particularly for Ekspress.

To calculate the performance that would be expected on real (unbalanced) data, we must take into account the expected real ratio of blocked to non-blocked comments. As Section 2 discusses, blocked comments are rarer than non-blocked, with the most recent estimate of the ratio from 2019 being 0.078 for 24sata. In practice, we would therefore expect for 24sata a recall of 0.67, a precision of 0.27 and an F-score of 0.38. In other words, the classifier would successfully detect 67% of comments that needed blocking (missing 33%), but 73% of its decisions to block would be false positives; and nearly 15% of innocent comments would be falsely blocked. While this level of performance is potentially useful, it seems it would still require significant manual filtering on the part of moderators. The balance between recall and precision could of course be tuned via the decision boundary, or by weighting the objective function in training, but gains in the recall would correspond to losses in precision, and vice versa (see Pavlopoulos et al., 2017a).

	24sata			Večernji List			Ekspress		
Model	ALL	BLK	NON	ALL	BLK	NON	ALL	BLK	NON
NB	69.43	47.59	91.26	66.39	49.75	81.79	64.57	46.48	82.66
LSTM	71.52	61.70	81.33	65.39	54.47	75.50	63.02	41.96	84.09
LASER	70.74	70.11	71.36	63.31	59.77	66.59	61.58	47.07	76.10
mBERT	76.42	67.33	85.49	69.63	53.18	84.87	68.40	58.46	78.34

Table 8: Classifier performance, as percentage accuracy. Columns are labelled ALL for all comments, BLK for positive instances only (blocked content), NON for negative instances only (non-blocked content).

Inspection of the conditional probability table produced by the NB model allows us to determine the words which are most strongly associated with the blocked and non-blocked classes, on the basis of the ratio of class probabilities. Tables 21 and 22 in Appendix B show full lists of the top 100 words for each class for 24sata. The strongest indicators for the blocked class correspond to vocabulary expected in spam comments: external URLs (*www*, *com*, *google*, *posjetite* (visit)); work and earnings (*poslu/posla* (work), *plaća* (payment), *zaradio/zaraditi* (earn)); amounts of money promised (numbers, *dolara* (dollars), *eura* (euros), *mjesecu* (monthly), *tjedno* (weekly), *dnevno* (daily)). Vocabulary associated with offensive language is also included, but comes further down the list (*jebem/jebo* (fuck), *majmun* (monkey)). Non-blocked indicators include vocabulary associated with discussion of a range of news topics (e.g. football: *inter*, *derbi*) and general evaluative words (*sretno/sritno* (happy/good luck), *predivno* (amazing), *najljepša* (most beautiful), *strašno* (terrible)). However, of a list of 185 blacklisted words used by the moderators at 24sata to flag comments for blocking, only 78 appear in the top 1000 in the NB model; and surprisingly, many words that one might expect to be associated with offensive or highly-charged language (although no blacklisted words) appear in the top 1000 non-blocked indicators in the NB model: *svastiku* (swastika), *terorizam* (terrorism), *trolaš* (you're trolling).

Vocabulary indicators extracted from these annotations are therefore not straightforward, suggesting that the data is fairly heterogeneous: comments may be blocked for many diverse reasons, and therefore display very different textual features. This may be one possible reason for the below-par performance; our next experiment investigates this.

4.3. Experiment 2: Blocking Rule Classification

For 24sata and VL, the publisher's database records the reason behind the moderators' decisions: the specific rule that a comment breaks. Here, we train and test multi-class versions of our classifier models for the problem of rule recognition.

		Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8
(a)	Train	24329	20	2167	30	2912	992	387	18786
	Val	3081	1	216	1	271	114	41	2457
	Test	2962	1	248	2	388	134	57	2444

		Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8	Rule9
(b)	Train	3652	6548	4514	57	3756	9	156	4	24322
	Val	572	794	547	7	402	0	13	0	2914
	Test	553	864	580	4	456	2	24	0	2951

Table 9: Blocking rule dataset distribution, for (a) 24sata and (b) Večernji List.

Table 9 shows the distribution of blocked comments by rule within the training, validation and test sets defined above. The distribution is very uneven: for 24sata, rules 1 (unrelated topics, spam, advertising etc.) and 8 (abuse, arguing with administrators) are common, while rules 2 (direct threats) and 4 (obscenity) are extremely rare; others are in between. For VL, rule 9 (using multiple accounts) is most common, with rules 4 (publishing personal data), 6 (using non-Latin script) and 8 (misinformation) very rare. Even for rules which seemingly map directly between the two schemata (e.g. hate speech: 24sata rule 3, VL rule 1; threats: 24sata ule 2, VL rule 2) the distributions seem to vary widely across newspapers: it seems to be very rare for 24sata moderators to class comments as threats, but quite common in VL.

One hypothesis might be that moderators tend to avoid applying rules with more serious consequences if other less serious ones could be used (see Tables 1 and 2); but while this might explain the rarity in 24sata of rules 2 (threats) and 4 (obscenity), it does not explain the distribution in VL, where rules 1 (hate speech), 2 (threats) and 9 (multiple accounts) are all commonly used. It may be that the ambiguity of many rules, together with the cultural practices and habits within particular groups of moderators, have significant effects here.

Results Table 10 shows the results for individual rules, with Table 11 showing the effect this would have on overall blocking accuracy (comments which break any rule should be blocked).

Performance for individual rules varies widely. Less frequent rules are often ignored by all classifiers (rules 2, 4), with better performance for more frequent rules (e.g. rules 1, 8). It is likely that the lower contribution of the less frequent classes to the training objective function means that not enough weight is given to them in the final classifier models. The NB model does much worse than other models, presumably because the pruning of the conditional probability table favours more common words, likely to be significant indicators of the more common classes. The simpler LSTM model seems to have an advantage over the more complex LASER and BERT models, in that accuracy seems more even across classes; this may be because the pre-training of the LASER and BERT models gives them less ability to adjust to the different classes in fine-tuning.

However, the overall performance is not strongly affected. Given the real blocking rate, for 24sata we would expect a recall of 0.48, a precision of 0.32 and an F-score of 0.39. This translates to successfully detecting 48% of comments that needed blocking (missing 52%), while producing 68% false positives; and blocking nearly 8% of innocent comments. Note that the F-score is very

(a)

Model	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8
NB	43.01	0	3.23	0	5.67	5.97	8.77	2.74
LSTM	62.42	0	56.05	0	50.52	75.37	43.86	57.53
LASER	51.25	0	9.68	0	1.55	16.42	0	50.12
mBERT	48.68	0	0	0	0	0	0	63.3

(b)

Model	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8	Rule9
NB	6.61	5.47	4.56	0	6.4	100	4.55	0	33.73
LSTM	25.73	20.64	33.65	50	35.22	0	13.64	0	40.41
LASER	51.39	45.26	67.49	66.67	61.37	0	63.64	0	57.85
mBERT	0	0	43.54	0	0	0	0	0	42.01

Table 10: Blocking rule classifier performance, measured as percentage accuracy, (a) 24sata (b) Večernji List.

Model	Overall	Blocked	Non-blocked
Chance	11.11	11.11	11.11
NB	60.06	22.19	97.93
LSTM	71.78	59.59	84.16
LASER	67.09	44.82	89.35
mBERT	70.04	47.93	92.19

Table 11: Performance of multi-class rule classifier on binary task, measured as percentage accuracy, 24sata.

similar to the classifier trained on the binary task, although the balance between precision and recall is different; this could be adjusted as discussed above.

To investigate the role of the multi-class objective function in training, we also checked the coverage of the classifiers trained on the binary task in Section 4.2 above. While these classifiers give only binary output and therefore cannot help moderators understand decisions, we can check how even their ability to detect the individual rules is. Table 12 shows the results. The very rarest classes (rules 2, 4) seem to behave quite randomly (given the very low counts, this is not surprising), but the slightly more common rules (6 and 7, then 3 and 5) get reasonable accuracy for most classifiers. The picture is mixed, however: some classes seem to be inherently hard to detect, with rules 5 (trolling) and 7 (non-Croatian language) getting relatively low scores for all classifiers.

Model	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8
NB	52.77	0	45.56	0	27.84	71.64	22.81	43.99
LSTM	63.37	100	61.29	50	52.84	79.85	56.14	60.27
LASER	71.0	100	69.76	100	58.25	84.33	42.11	70.79
mBERT	64.15	0	72.18	100	54.64	88.06	36.84	72.3

Table 12: Performance of binary classifier per blocking rule, measured as percentage accuracy, 24sata.

4.4. Experiment 3: Variation over Time

Another possible reason for variable performance is the reliability and/or variability of the moderation annotation itself. Moderation can be quite a subjective decision, and the large amounts of data to mean that many blockable comments may be missed. One way to test this is to examine how classifier performance changes over time, as the moderation policy and the amount of effort put into moderation changed over the years (see Section 4.2.1); for this experiment we focus on just one dataset, 24sata. The distribution of individual blocking rules also varies over time: Figure 2 shows the proportion of blocking decisions based on each rule for the last four years (the years with most data). (Full details of the rule distributions over time for both 24sata and VL are given in Appendix A, Section A.2). Significant changes can be seen in the proportions: it seems that the most common classes (rules 1 and 8) become less used over time, with rarer classes increasing. It therefore seems likely that rules are being applied differently in different cases: with many rules covering a range of phenomena and many phenomena being covered by multiple rules (see details of the rules in Table 1 above), moderators have a choice in which rules to apply, and perhaps more specific rules (often with more stringent penalties) are becoming preferred.

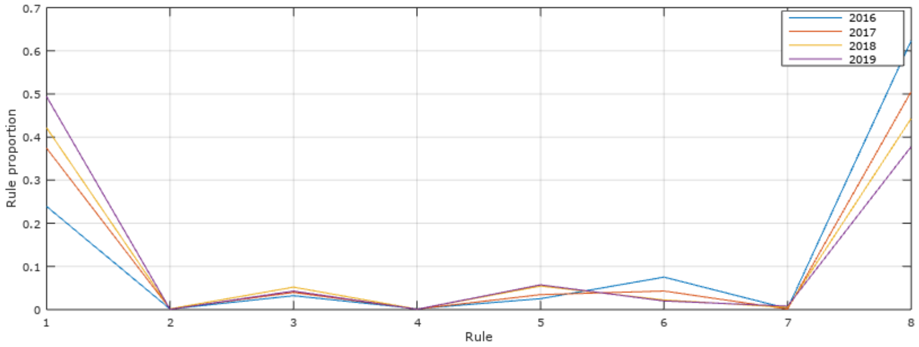


Figure 2: Blocking rule proportion over time, 24sata.

To determine the variability of the models' performance over different years' data, we therefore created a series of test sets, one for each of the last four years. We keep the same training set, taken from 2019 data (see above); the 2019 test set is therefore smaller and based on that used in the previous section. The test sets for 2016-2018 are larger as they can contain all the year's data labelled with rules; as the training set is fixed we can also test on a realistic balance of data, using all the blocked and non-blocked comments available for each year. Table 13 shows the test set distribution over time.

Results Table 14 shows overall accuracy figures per year on the 24sata dataset; we show only performance for the best classifier model, mBERT. Accuracy decreases as we move further away from the year 2019 used in training. Table 15 then shows how the accuracy of the binary blocking classifier varies with blocking rule class: while figures for many rules decrease in years before 2019, performance for rules 3 (hate speech), 6 (vulgarity) and perhaps 8 (abuse of other users, authors and admins) seems to remain relatively steady. Performance for rule 2 (threats) and rule

	Articles	Non-blocked	Blocked	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8
2016	907	196762	15154	2915	111	992	183	683	1413	227	8630
2017	1045	188639	20579	6351	185	1560	153	1273	1211	137	9709
2018	1678	285620	21838	237	254	2800	125	2616	840	780	14186
2019	1154	68706	6398	3070	3	256	2	396	138	58	2475

Table 13: Yearwise dataset distribution, 24sata.

Year	Overall	Blocked	Non-blocked	F1-macro	Recall (BLK)	Precision (BLK)
2016	72.25	72.20	72.89	54.19	0.73	0.15
2017	75.17	76.16	64.84	58.10	0.65	0.21
2018	76.75	78.36	61.32	59.59	0.61	0.23
2019	80.03	81.19	67.32	62.07	0.67	0.25

Table 14: Binary classification performance over the yearwise testset using mBERT, 24sata. Figures are shown as percentage accuracy overall and for the blocked and non-blocked content separately; as this experiment uses the full data for each year (rather than a balanced subset) we also give F1 score macro-averaged over the two classes, and recall and precision for the blocked class only.

7 (non-Croatian language) may even be improving, although these rules have smaller amounts of data. Some of the main categories that relate to offensive language therefore seem to remain relatively consistent, while other categories such as advertising, spam and distribution of obscene content may be changing more. This may be because topics and vocabulary change over time; because authors change their language to avoid detection; because moderators change their criteria and behaviour; or a combination of these factors. What seems clear is that change over time is a significant issue: the ability to re-train classifiers on new data and up-to-date moderation labels will be important in practice.

5. Discussion and Conclusions

In this section we discuss the possible reasons for the overall levels of performance observed, and draw conclusions about what steps can be taken to improve it.

Year	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8
2016	52.37	65.00	75.85	46.07	46.77	93.51	63.96	78.62
2017	49.36	76.92	70.27	51.68	46.99	85.71	71.21	73.34
2018	50.67	83.54	71.74	42.74	37.74	90.93	38.20	68.73
2019	64.23	66.67	72.18	100.00	54.36	88.32	35.85	72.17

Table 15: Blocking rule classification performance over the yearwise testset using mBERT, measured as percentage accuracy, 24sata.

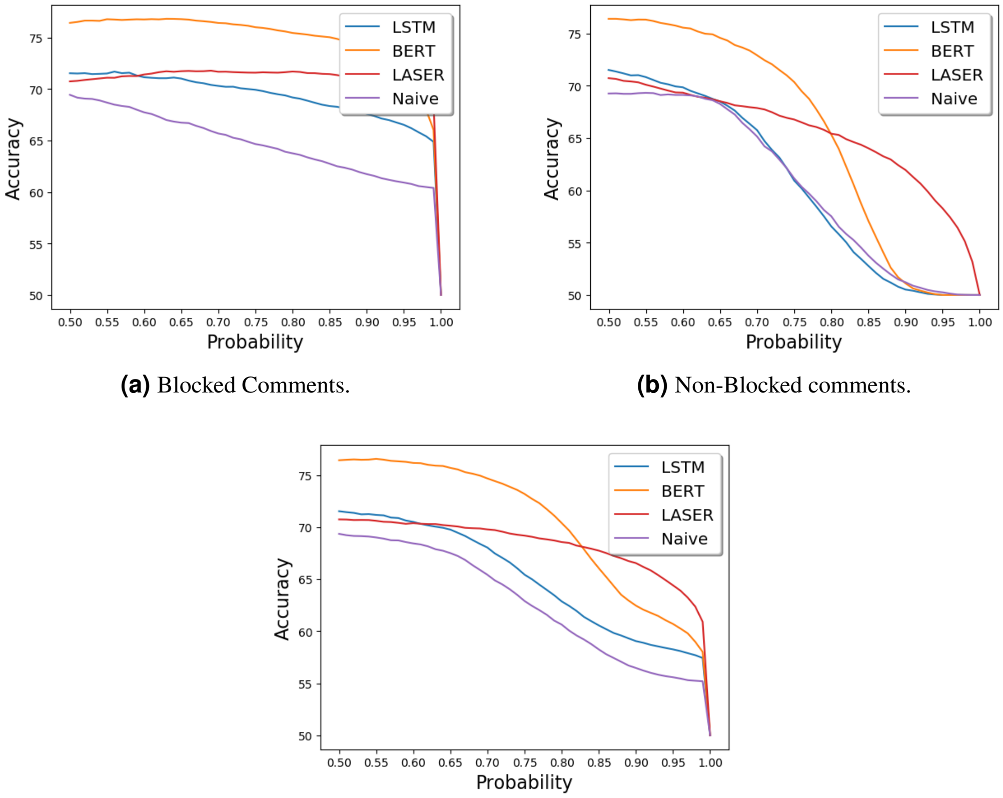
5.1. Analysis of Classifier Outputs

Figure 3 shows the confidence of the different classifier models: the plots are generated by changing the decision threshold of each classifier, increasing from the default 0.5 up to 1.0, and calculating the classification accuracy on the standard 24sata test set of Section 4.2. This is shown for blocked comments in Figure 3a, for non-blocked comments in Figure 3b, and the overall average in Figure 3c. The BERT and LASER models show overall higher confidence: increasing the threshold at which the decision is made has less effect on the accuracy of their output. The NB and to a lesser extent LSTM models' performance drops off more quickly, showing that their outputs give lower confidences for many correct classification decisions. Interestingly, classifier confidences seem significantly higher for blocked comments: the dropoff in performance is much less than that for non-blocked comments as the threshold increases. Although its performance was generally lower, the LASER model may provide some advantages here: its confidence curve is flatter with less dropoff for non-blocked comments.

This general tendency suggests that non-blocked comments are harder to classify in many cases. This may be due to variability or lack of reliability in moderation, with many comments that should be blocked labelled as non-blocked. Classifiers would therefore be learning decision boundaries that fit these examples where possible, but having to leave them close to the boundary given their similarity to other blocked comments.

Manual inspection of classifier errors was carried out over a set of approximately 350 comments on which the best (mBERT) classifier output disagreed with the moderator's decisions. These comments were passed back to 24sata's moderators, who were asked to moderate them again and produce a new set of labels. Of 101 comments which were originally not blocked, the majority (82) were still not blocked, but with a significant proportion (19) now marked as blocked. The problem of moderators missing comments which should be blocked is therefore a real one, as suspected. However, a bigger effect may be the variability of moderation decisions. Of 244 comments which were originally blocked (but given a non-blocked label by our classifier), approximately half (124) were still judged to be blocked, but half (120) were now marked non-blocked. Of the 124 which remained blocked, over half (81) were given a different rule as justification for blocking.

Examination of the errors also helps shed some light on the phenomena which cause difficulties for automatic classification. Some examples show classic language processing problems: non-standard spelling and vocabulary, and complex references and indirect statements can all be hard for classifiers to recognise without extremely large training sets. Two particular phenomena emerge as covering a large proportion of examples, however. One is that reader comments occur in the context of the article and the preceding comments, and many references need that context to be understood (see example (1), in which the phrase "that symbol" refers to an important concept from the previous discussion, probably the swastika. Treating comments as independent texts (as we do here) misses this – without the reference, it is hard to understand the comment as problematic. The second is that many comments use culture- and country-specific references which must also be resolved before the stance of the comment is clear. Example (2) appears on the face of it as a political trolling attempt; but if one knows that the HDZ and SDP are not only opposing political parties, but the only two large parties in Croatia, it can be understood as even-handed. In example (3), one must know that Pavelić headed a fascist government, and that



(a) Blocked Comments.

(b) Non-Blocked comments.

(c) Both comments.

Figure 3: Confidence of the Classifier.

Tuđman founded the currently governing, right-of-centre HDZ, in order to see its provocative nature.

(1) U čemu je problem? Dotični je pod tim simbolom živio i djelovao.
What's the problem? The person in question lived and worked under that symbol.
Moderator decision: blocked, rule 8

(2) HDZ je proslost a i Sdp !
HDZ is the past, and so is the SDP!
Moderator decision: not blocked

(3) Naime, preko natpisa "Franjo Tuđman, prvi hrvatski predsjednik"... Profesor Milan Kangrga je u emisiji NU2 rekao da je prvi hr pred bio Ante Pavelić.
Namely, via the inscription "Franjo Tuđman, the first Croatian president" ... Professor Milan Kangrga said on the NU2 show that the first Croatian president was Ante Pavelić.
Moderator decision: blocked, rule 8/rule 5 (moderators disagree)

5.2. Conclusions and Further Work

The high levels of variability in moderation decisions, and in the justifications given for them according to the moderation policy, indicate that an iterative approach may be of benefit in this task. Working with moderators to jointly define a more reliable policy, based partly on observation and use of high-confidence classifier outputs as in the error analysis above, would allow us to work towards less noisy data together with more reliable and useful classifiers. This could be framed within a general active learning approach, and we hope to explore this in future work. However, working within a real-world setting constrains the time and resources that can be dedicated to such work; great care must be taken to find an approach which does not further burden moderators and news publishers.

Second, the use of moderation flags as training labels, as pursued here and in other related work (Pavlopoulos et al., 2017a), may not be the most practical way to proceed in order to produce an accurate classification tool. A more effective and reliable way may be to use other, better-understood and curated datasets which represent the categories of language and author behaviour which should be blocked. By training classifiers on these cleaner datasets, a more reliable set of classifier outputs may be obtained which can feed into an active learning approach as outlined above. However, as Section 3 explains, such datasets are simply not available in the languages of interest here (Croatian and Estonian), or in many other language other than the majority well-resourced languages such as English, German and Spanish. One helpful step might be to pre-train word embeddings and/or models on data in the target language, even if annotated data is not available, to help smooth the noise from the training set; but note that the LASER and BERT models used here already benefit from large amounts of multi-lingual data, and in any case this is unlikely to go far towards solving the problem. Cross-lingual approaches (Ruder et al., 2017) would therefore be of great benefit if they can permit transfer learning from well-understood datasets in better-resourced languages to tasks in less-resourced languages.

However, while some work in hate speech and offensive language detection has been multilingual, studying datasets in more than one language, cross-lingual work is rare. A. Basile & Rubagotti (2018) use a *bleaching* approach (van der Goot et al., 2018) to conduct cross-lingual experiments between Italian and English in the EVALITA 2018 misogyny identification task, and Pamungkas & Patti (2019) propose a cross-lingual approach using a LSTM joint-learning model with multilingual MUSE embeddings. However, as far as we are aware, no work has yet tried to apply this to the problem of comment filtering, or focused on the languages needed here. As our error analysis shows, the task here poses significant challenges for cross-lingual techniques: many phenomena of interest are dependent on region- or culture-specific references and understanding of the related context, as in the need to understand country-specific relations between political parties and individuals discussed in the previous section. Current cross-lingual techniques depend on parallel corpus training, or on mapping of embedding spaces based on known synonymous anchor points (e.g. digits); these are unlikely to capture such phenomena well. Our next steps will therefore be to adapt techniques for cross-lingual learning to try to better map the entities, events and similar references found in news text between languages.

6. Acknowledgements

This research is supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 825153, project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media). The results of this publication reflect only the authors' views and the Commission is not responsible for any use that may be made of the information it contains.

References

Aiyar, S., & Shetty, N. P. (2018). N-gram assisted Youtube spam comment detection. *Procedia Computer Science*, *132*, 174 - 182. Retrieved from http://www.sciencedirect.com/science/article/pii/S1877050918309153 (International Conference on Computational Intelligence and Data Science) doi: https://doi.org/10.1016/j.procs.2018.05.181

Alberto, T., Lochter, J., & Almeida, T. (2015, December). TubeSpam: Comment spam filtering on YouTube. In *Proceedings of the 14th ieee international conference on machine learning and applications (icmla'15)* (p. 1-6).

Álvarez-Carmona, M. Á., Guzmán-Falcón, E., Montes-y Gómez, M., Escalante, H. J., Villasenor-Pineda, L., Reyes-Meza, V., & Rico-Sulayes, A. (2018). Overview of MEX-A3T at IberEval 2018: Authorship and aggressiveness analysis in Mexican Spanish tweets. In *Notebook papers of 3rd sepln workshop on evaluation of human language technologies for iberian languages (ibereval), seville, spain* (Vol. 6).

Anzovino, M., Fersini, E., & Rosso, P. (2018). Automatic identification and classification of misogynistic language on Twitter. In M. Silberztein, F. Atigui, E. Kornyshova, E. Métais, & F. Meziane (Eds.), *Natural language processing and information systems (NLDB)* (Vol. 10859, p. 57-64). Springer.

Artetxe, M., & Schwenk, H. (2019). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, *7*, 597–610.

Badawy, A., Ferrara, E., & Lerman, K. (2018, Aug). Analyzing the digital traces of political manipulation: The 2016 Russian interference Twitter campaign. In *Ieee/acm international conference on advances in social networks analysis and mining (asonam)* (p. 258-265). doi: 10.1109/ASONAM.2018.8508646

Barker, E., Paramita, M. L., Aker, A., Kurtic, E., Hepple, M., & Gaizauskas, R. (2016, September). The SENSEI annotated corpus: Human summaries of reader comment conversations in on-line news. In *Proceedings of the 17th annual meeting of the special interest group on discourse and dialogue* (pp. 42–52). Los Angeles: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/W16-3605 doi: 10.18653/v1/W16-3605

Basile, A., & Rubagotti, C. (2018). Crotonemilano for ami at evalita2018. a performant, cross-lingual misogyny detection system. In *Evalita@ clic-it.*

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rangel Pardo, F. M., … Sanguinetti, M. (2019, June). SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proc. semeval* (pp. 54–63). Retrieved from `https://www.aclweb.org/anthology/S19-2007` doi: 10.18653/v1/S19-2007

Caruana, G., & Li, M. (2012, March). A survey of emerging approaches to spam filtering. *ACM Computing Surveys*, *44*(2), 9:1–9:27. Retrieved from `http://doi.acm.org/10.1145/2089125.2089129` doi: 10.1145/2089125.2089129

Chen, C., Zhang, J., Chen, X., Xiang, Y., & Zhou, W. (2015, June). 6 million spam tweets: A large ground truth for timely Twitter spam detection. In *2015 ieee international conference on communications (icc)* (p. 7065-7070). doi: 10.1109/ICC.2015.7249453

de Gibert, O., Perez, N., García-Pablos, A., & Cuadros, M. (2018, October). Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd workshop on abusive language online (ALW2)* (pp. 11–20). Brussels, Belgium: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/W18-5102` doi: 10.18653/v1/W18-5102

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805.*

Ghosh, S., Viswanath, B., Kooti, F., Sharma, N. K., Gautam, K., Benevenuto, F., … Gummadi, K. (2012, April). Understanding and Combating Link Farming in the Twitter Social Network. In *Proceedings of the 21st International World Wide Web Conference (WWW'12).* Lyon, France.

Hochreiter, S., & Schmidhuber, J. (2015). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780.

Jurafsky, D., & Martin, J. (2009). *Speech and language processing* (2nd ed.). Pearson Prentice Hall.

Kantchelian, A., Ma, J., Huang, L., Afroz, S., Joseph, A. D., & Tygar, J. D. (2012, October). Robust detection of comment spam using entropy rate. In *Proceedings of the 5th acm workshop on artificial intelligence and security* (p. 59-70).

Kim, D., Graham, T., Wan, Z., & Rizoiu, M. (2019). Tracking the digital traces of Russian trolls: Distinguishing the roles and strategy of trolls on Twitter. *CoRR*, *abs/1901.05228*. Retrieved from `http://arxiv.org/abs/1901.05228`

Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd international conference on learning representations (iclr).*

Kolhatkar, V., Wu, H., Cavasso, L., Francis, E., Shukla, K., & Taboada, M. (2019, Nov 02). The sfu opinion and comments corpus: A corpus for the analysis of online news comments. *Corpus Pragmatics*. Retrieved from https://doi.org/10.1007/s41701-019-00065-w doi: 10.1007/s41701-019-00065-w

Linvill, D. L., & Warren, P. L. (2018). *Troll factories: The internet research agency and state-sponsored agenda building.* Retrieved from http://pwarren.people.clemson.edu/Linvill_Warren_TrollFactory.pdf

Liu, P., Guberman, J., Hemphill, L., & Culotta, A. (2018). Forecasting the presence and intensity of hostility on instagram using linguistic and social features. In *Twelfth international aaai conference on web and social media.*

Ljubešić, N., Erjavec, T., & Fišer, D. (2018, October). Datasets of Slovene and Croatian moderated news comments. In *Proc. 2nd workshop on abusive language online* (pp. 124–131). Retrieved from https://www.aclweb.org/anthology/W18-5116 doi: 10.18653/v1/W18-5116

Ljubešić, N., Fišer, D., & Erjavec, T. (2019). The FRENK datasets of socially unacceptable discourse in Slovene and English. *CoRR*, *abs/1906.02045*. Retrieved from http://arxiv.org/abs/1906.02045

MacAvaney, S., Yao, H.-R., Yang, E., Russell, K., Goharian, N., & Frieder, O. (2019, 08). Hate speech detection: Challenges and solutions. *PLOS ONE*, *14*(8), 1-16. Retrieved from https://doi.org/10.1371/journal.pone.0221152 doi: 10.1371/journal.pone.0221152

Mihaylov, T., & Nakov, P. (2016, August). Hunting for troll comments in news community forums. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 399–405). Berlin, Germany: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/P16-2065 doi: 10.18653/v1/P16-2065

Mojica, L. G. (2017). A trolling hierarchy in social media and A conditional random field for trolling detection. *CoRR*, *abs/1704.02385*. Retrieved from http://arxiv.org/abs/1704.02385

Mojica de la Vega, L. G., & Ng, V. (2018, May). Modeling trolling in social media conversations. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC-2018).* Miyazaki, Japan: European Languages Resources Association (ELRA). Retrieved from https://www.aclweb.org/anthology/L18-1585

Napoles, C., Tetreault, J., Rosata, E., Provenzale, B., & Pappu, A. (2017, April). Finding good conversations online: The yahoo news annotated comments corpus. In *Proceedings of the 11th linguistic annotation workshop* (pp. 13–23). Valencia, Spain: Association for Computational Linguistics.

Narayanan, A. (2018). *Tweets dataset for detection of cyber-trolls.* Retrieved from `https://www.kaggle.com/dataturks/dataset-for-detection-of-cybertrolls`

Ousidhoum, N., Lin, Z., Zhang, H., Song, Y., & Yeung, D.-Y. (2019). Multilingual and multi-aspect hate speech analysis. *arXiv preprint arXiv:1908.11049.*

Pamungkas, E. W., & Patti, V. (2019). Cross-domain and cross-lingual abusive language detection: A hybrid approach with deep learning and a multilingual lexicon. In *Proceedings of the 57th annual meeting of the association for computational linguistics: Student research workshop* (pp. 363–370).

Pavlopoulos, J., Malakasiotis, P., & Androutsopoulos, I. (2017a, September). Deeper attention to abusive user content moderation. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 1125–1135). Copenhagen, Denmark: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/D17-1117` doi: 10.18653/v1/D17-1117

Pavlopoulos, J., Malakasiotis, P., & Androutsopoulos, I. (2017b, August). Deep learning for user comment moderation. In *Proceedings of the first workshop on abusive language online* (pp. 25–35). Vancouver, BC, Canada: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/W17-3004` doi: 10.18653/v1/W17-3004

Ruder, S., Vulić, I., & Søgaard, A. (2017). A survey of cross-lingual word embedding models. *CoRR*, *abs/1706.04902*. Retrieved from `http://arxiv.org/abs/1706.04902`

Schmidt, A., & Wiegand, M. (2017, April). A survey on hate speech detection using natural language processing. In *Proceedings of the 5th international workshop on natural language processing for social media* (pp. 1–10). Retrieved from `https://www.aclweb.org/anthology/W17-1101` doi: 10.18653/v1/W17-1101

van der Goot, R., Ljubešić, N., Matroos, I., Nissim, M., & Plank, B. (2018). Bleaching text: Abstract features for cross-lingual gender prediction. *arXiv preprint arXiv:1805.03122.*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (p. 5998-6008).

Waseem, Z., & Hovy, D. (2016, 01). Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In *Proceedings of the naacl student research workshop* (p. 88-93).

Wiegand, M., Siegel, M., & Ruppenhofer, J. (2018, September). Overview of the GermEval 2018 shared task on the identification of offensive language. In *Proceedings of GermEval 2018, 14th conference on natural language processing (KONVENS).* Vienna, Austria.

Wu, T., Wen, S., Xiang, Y., & Zhou, W. (2018). Twitter spam detection: Survey of new approaches and comparative study. *Computers and Security*, *76*, 265-284. Retrieved from `http://www.sciencedirect.com/science/article/pii/S016740481730250X` doi: https://doi.org/10.1016/j.cose.2017.11.013

Wulczyn, E., Thain, N., & Dixon, L. (2017). Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th international conference on world wide web* (pp. 1391–1399).

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019). Semeval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). *CoRR*, *abs/1903.08983*. Retrieved from `http://arxiv.org/abs/1903.08983`

Zhang, J., Chang, J., Danescu-Niculescu-Mizil, C., Dixon, L., Hua, Y., Taraborelli, D., & Thain, N. (2018, July). Conversations gone awry: Detecting early signs of conversational failure. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1350–1361). Melbourne, Australia: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/P18-1125` doi: 10.18653/v1/P18-1125

A. Yearwise Data Distribution

This section gives the full details of the dataset distributions over time, in terms of overall numbers of articles, comments and moderator's blocking behaviour for all three newspapers (Section 4.2.1), and the frequency of application of individual blocking rules for 24sata and VL (Section 4.4).

A.1. Summary data, commenting and blocking rates

Year	# Articles	# Comments	# Blocked	Comment rate	Blocking rate
2007	6054	38005	3	6.3	7.9×10^{-5}
2008	26523	185578	12	7.0	6.5×10^{-5}
2009	38024	326609	31	8.6	9.5×10^{-5}
2010	38777	459227	2	11.8	4.4×10^{-6}
2011	38330	1140555	111	29.8	9.7×10^{-5}
2012	43978	1870449	251	42.5	1.3×10^{-4}
2013	46457	2490285	130	53.6	5.2×10^{-5}
2014	46429	2656841	171	57.2	6.4×10^{-5}
2015	44919	3054087	724	68.0	2.4×10^{-4}
2016	47595	3194761	98487	67.1	3.1×10^{-2}
2017	45891	2795824	134080	60.9	4.8×10^{-2}
2018	48777	2519279	156083	51.7	6.2×10^{-2}
2019	17953	816692	63972	45.5	7.8×10^{-2}
Total	489707	21548192	454057		

Table 16: Yearwise data distribution, 24sata; comment rate = $N_{\text{comments}}/N_{\text{articles}}$, blocking rate = $N_{\text{blocked}}/N_{\text{comments}}$.

Year	# Articles	# Comments	# Blocked	Comment rate	Blocking rate
2009	7724	162017	4	20.98	2.47×10^{-5}
2010	31423	764134	175	24.32	$\times10^{-4}$
2011	32521	1245946	91	38.31	7.30×10^{-5}
2012	35693	1022186	29	28.64	2.84×10^{-5}
2013	41408	1101234	16747	26.59	1.52×10^{-2}
2014	43251	835152	48099	19.31	5.76×10^{-2}
2015	43469	1237714	48930	28.47	3.95×10^{-2}
2016	40485	1009070	60390	24.92	5.98×10^{-2}
2017	38136	840677	87476	22.04	1.04×10^{-1}
2018	42092	1073953	130054	25.51	1.21×10^{-1}
2019	16453	354551	55295	21.55	1.56×10^{-1}
Total	372655	9646634	447290		

Table 17: Yearwise data distribution, Večernji List; comment rate $= N_{\text{comments}}/N_{\text{articles}}$, blocking rate $= N_{\text{blocked}}/N_{\text{comments}}$.

Year	# Articles	# Comments	# Blocked	Comment rate	Blocking rate
2009	109352	2898438	130040	26.51	4.49×10^{-2}
2010	105173	2377591	107735	22.61	4.53×10^{-2}
2011	127037	2729389	148302	21.49	5.43×10^{-2}
2012	127663	3372776	249880	26.42	7.41×10^{-2}
2013	114914	3289393	295608	28.63	8.99×10^{-2}
2014	101936	3195502	336450	31.35	10.53×10^{-2}
2015	98198	3202592	391758	32.61	12.23×10^{-2}
2016	94353	2848624	355868	30.19	12.49×10^{-2}
2017	87098	2838075	265810	32.58	9.37×10^{-2}
2018	82887	3194597	343538	38.54	10.75×10^{-2}
2019	32691	1540382	188197	47.12	12.21×10^{-2}
Total	1081302	31487359	2813186		

Table 18: Yearwise data distribution, Ekspress; comment rate $= N_{\text{comments}}/N_{\text{articles}}$, blocking rate $= N_{\text{blocked}}/N_{\text{comments}}$.

A.2. Blocking rule distribution

	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8
2007						1		2
2008	12							
2009	29		1					1
2010	2							
2011	107							4
2012	144				2	9	13	83
2013	112				5		1	12
2014	108	1	1		45	2		14
2015	659	2	7		18	1		37
2016	23551	111	3152	183	2479	7400	227	61384
2017	50178	185	5310	153	4631	5752	137	67734
2018	65775	254	8099	125	8483	3453	780	69114
2019	31592	26	2734	37	3658	1270	498	24157

Table 19: Yearwise blocking rule data distribution, 24sata.

	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8	Rule9
2009		4							
2010	91		1		1				82
2011	44		1		4				42
2012	8		4	2	4				11
2013	1748	52	6575	16	7192	15	618	275	256
2014	4913	83	20911	19	16462	114	813	142	4642
2015	5438	82	16729	24	21858	109	187	4	4499
2016	4859	118	14007	10	38076	147	889	2	2282
2017	28888	169	15251	30	35957	195	608		6378
2018	33660	8076	17311	4	37572	45	256	8	33122
2019	4860	8477	5748	72	4712	11	199	4	31212

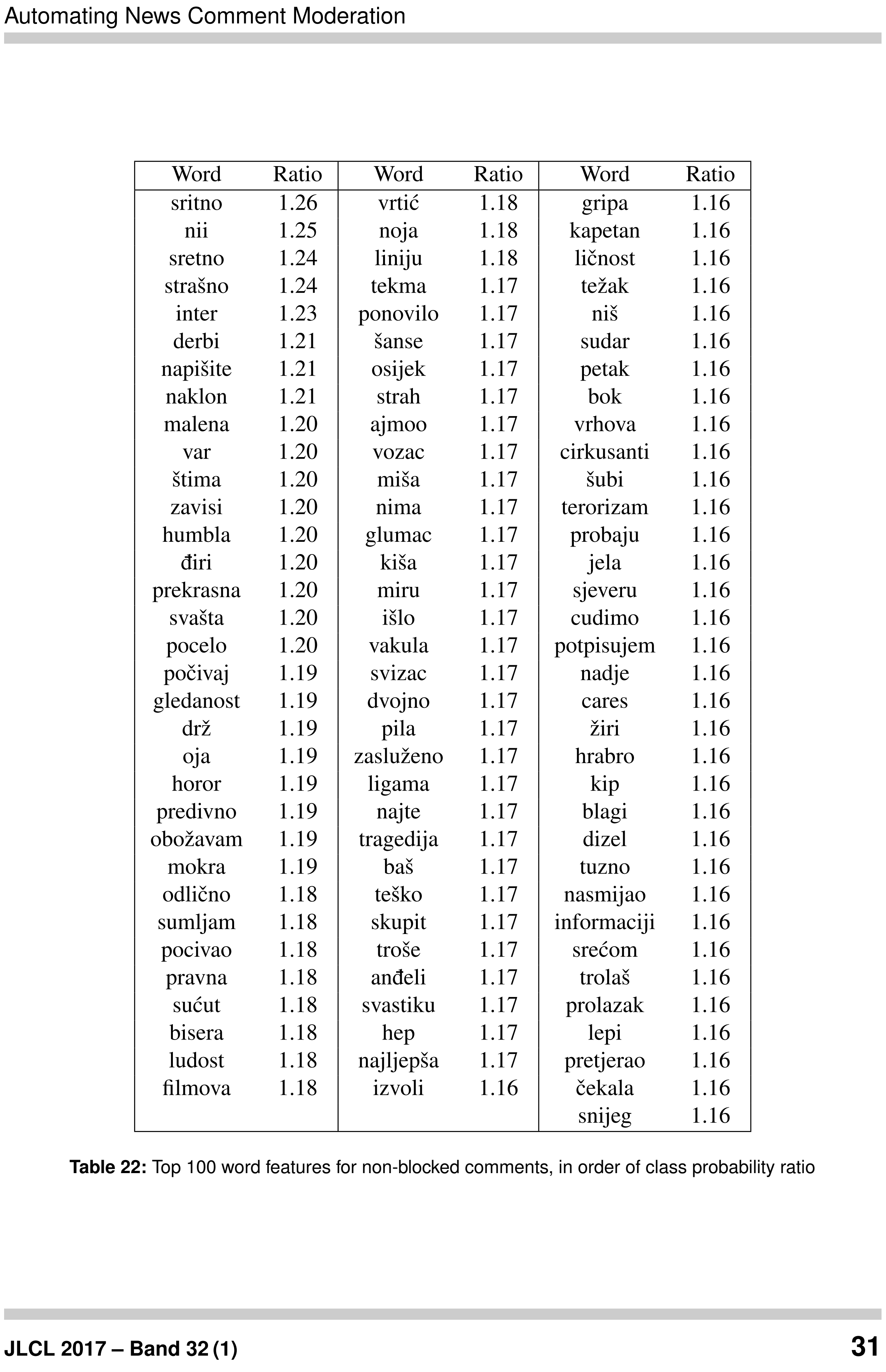
Table 20: Yearwise blocking rule data distribution, Večernji List.

B. Word Lists

This section gives the full top 100 words lists for blocked and non-blocked comments as inferred by the Naïve Bayes classifier trained on the binary classification task (Section 4.2.2).

Word	Ratio	Word	Ratio	Word	Ratio
20544	2.16	ponio	1.50	jebo	1.40
22000	2.14	ovom	1.48	odnio	1.40
17000	2.14	ovog	1.48	ženu	1.40
pridružio	2.08	želiš	1.47	sada	1.40
mreži	2.08	internetu	1.47	dobivanje	1.40
www	2.03	radno	1.46	nepunim	1.39
com	2.02	jebem	1.46	redoviti	1.39
mjesečno	1.94	promijenilo	1.45	pogledam	1.39
google	1.94	slijedite	1.45	radeci	1.39
mjesecu	1.85	dnevno	1.45	sponzoru	1.39
kuće	1.81	paycheck	1.44	šokiran	1.38
dolara	1.80	eura	1.44	redovne	1.38
mjeseca	1.79	odlučio	1.44	počeo	1.38
prvom	1.78	dnevne	1.44	stanicom	1.38
poslu	1.77	nabijem	1.43	odabirete	1.38
zaradio	1.76	litte	1.43	primio	1.38
rad	1.74	24857	1.43	vremenom	1.37
radeći	1.70	čula	1.43	zarađivati	1.37
promijenjen	1.69	web	1.42	želite	1.36
plaća	1.69	top	1.42	blogu	1.36
dobrodošli	1.69	započela	1.42	prije	1.36
7645	1.67	premise	1.42	dodatni	1.36
9264	1.67	rasponu	1.42	86	1.36
27936	1.67	prošlog	1.42	prethodni	1.36
tjedno	1.57	počinjem	1.41	zaradite	1.35
online	1.57	četiri	1.41	rate	1.35
pronaći	1.55	jednostavan	1.41	39	1.35
mom	1.54	29584	1.41	stranicu	1.35
posla	1.53	22738	1.41	posjetite	1.35
zaraditi	1.53	sam	1.41	majmune	1.35
noć	1.52	debil	1.40	mijenjam	1.34
skraćeno	1.52	računalo	1.40	govno	1.34
satu	1.51	jo	1.40	nepuno	1.34
				mjesec	1.34

Table 21: Top 100 word features for blocked comments, in order of class probability ratio



Word	Ratio	Word	Ratio	Word	Ratio
sritno	1.26	vrtić	1.18	gripa	1.16
nii	1.25	noja	1.18	kapetan	1.16
sretno	1.24	liniju	1.18	ličnost	1.16
strašno	1.24	tekma	1.17	težak	1.16
inter	1.23	ponovilo	1.17	niš	1.16
derbi	1.21	šanse	1.17	sudar	1.16
napišite	1.21	osijek	1.17	petak	1.16
naklon	1.21	strah	1.17	bok	1.16
malena	1.20	ajmoo	1.17	vrhova	1.16
var	1.20	vozac	1.17	cirkusanti	1.16
štima	1.20	miša	1.17	šubi	1.16
zavisi	1.20	nima	1.17	terorizam	1.16
humbla	1.20	glumac	1.17	probaju	1.16
điri	1.20	kiša	1.17	jela	1.16
prekrasna	1.20	miru	1.17	sjeveru	1.16
svašta	1.20	išlo	1.17	cudimo	1.16
pocelo	1.20	vakula	1.17	potpisujem	1.16
počivaj	1.19	svizac	1.17	nadje	1.16
gledanost	1.19	dvojno	1.17	cares	1.16
drž	1.19	pila	1.17	žiri	1.16
oja	1.19	zasluženo	1.17	hrabro	1.16
horor	1.19	ligama	1.17	kip	1.16
predivno	1.19	najte	1.17	blagi	1.16
obožavam	1.19	tragedija	1.17	dizel	1.16
mokra	1.19	baš	1.17	tuzno	1.16
odlično	1.18	teško	1.17	nasmijao	1.16
sumljam	1.18	skupit	1.17	informaciji	1.16
pocivao	1.18	troše	1.17	srećom	1.16
pravna	1.18	anđeli	1.17	trolaš	1.16
sućut	1.18	svastiku	1.17	prolazak	1.16
bisera	1.18	hep	1.17	lepi	1.16
ludost	1.18	najljepša	1.17	pretjerao	1.16
filmova	1.18	izvoli	1.16	čekala	1.16
				snijeg	1.16

Table 22: Top 100 word features for non-blocked comments, in order of class probability ratio