Approaching Automatic Cyberbullying Detection for Polish Tweets

Krzysztof Wróbel (Department of Computational Linguistics, Jagiellonian University)

Abstract

This paper presents contribution to PolEval 2019[1] automatic cyberbullying detection task. The goal of the task is to classify tweets as harmful or normal. Firstly, the data is preprocessed. Then two classifiers adjusted to the problem are tested: Flair and fastText. Flair utilizes character-based language models, which are evaluated using perplexity. Both classifiers obtained similar scores on test data.

Keywords

natural language processing, text classification, cyberbullying detection, Polish

1. Introduction

The problem of automatic cyberbullying detection for Polish have been introduced in PolEval 2019 contest. The goal of the task is to classify user-generated content as harmful or non-harmful. The cyberbullying problem has a real impact on people lives, even suicides were committed because of it.

Ptaszyński et al. (2018) evaluated a couple of cyberbullying detection systems. The best results were obtained by a proprietary system, but the second was fastText classifier beating many commercial systems.

In this paper, two classifiers adjusted to the problem are evaluated. The results were submitted to PolEval contest.

[1] `http://poleval.pl`

2. Data

Data provided by organizers consists of Polish tweets annotated as harmful and non-harmful (normal). In Subtask 2 harmful class is divided into cyberbullying and hate-speech. In training and test data, all user mentions are anonymized and shared tweets (beginning with `RT`) are truncated (the truncated tweets ends with ellipsis Unicode character). Last characters or words of tweets may carry important information for a classifier, e.g. emoticons.

Table 1 shows the distribution of classes in the training and test data. The number of harmful tweets is about 10 times smaller than normal tweets. Distribution of classes in training and test data does not match exactly.

Table 1: Distribution of classes in training and test data for Subtasks 1 and 2

Class	Subtask 1		Subtask 2	
	Training	Testing	Training	Testing
0	9190 (91.52%)	866 (86.60%)	9190 (91.52%)	866 (86.60%)
1	851 (8.48%)	134 (13.40%)	253 (2.52%)	25 (2.50%)
2	–	–	598 (5.96%)	109 (10.90%)

In order to employ a more suitable language model, new tweets were collected for 3 days using Twitter Streaming API. In comparison to the PolEval data, new tweets are full text and are not anonymized. The corpus (referenced later as the raw Twitter corpus) consists of 1.7 millions of tweets (164 MB of raw text).

3. Approach

Firstly, the data was preprocessed:

— frequent emojis were replaced to ASCII versions, e.g. smiling face was replaced to `:)`

— beginning retweet mark (`RT`) was removed

— escaped new line (`\n`) was replaced to space

— escaped quotation mark (`\"`) was unescaped

— encoded ampersand (`\u0026`) was replaced to ampersand (`&`).

For text classification two libraries were employed: Flair (Akbik et al. 2018) with addressed imbalance and fastText (Bojanowski et al. 2017, Joulin et al. 2017) using pretrained embeddings.

3.1. Flair

Flair generates contextual embeddings for a span of text (e.g. word) using character-based language models: forward and backward. Language models are trained on raw corpora. Table 2 shows character perplexity using language models trained on National Corpus of Polish (NKJP; Pęzik 2012), KGR10 (Kocoń and Gawor 2018), the raw Twitter corpus, and Common Crawl. The perplexity was calculated on training and test data on the original form and without anonymized mentions (`@anonymized_account`), and on a separate fragment of the raw Twitter corpus. The raw Twitter corpus was left unprocessed (i.e. not processed the same as the task data).

Table 2: Character perplexity of language models trained on different corpora tested on the original PolEval data, PolEval data without user mentions, and a fragment of the raw Twitter corpus

Corpus	Original data		Without anonymized mentions		Twitter corpus
	Training	Testing	Training	Testing	
NKJP	7 806	7 840	4 903	4 789	7 474
KGR10	**6 614**	**6 568**	**4 705**	**4 549**	6 870
Twitter	7 826	7 941	4 725	4 709	**3 396**
Common Crawl	11 820	12 025	6 678	6 714	10 564

Language model trained on Twitter corpus has a significantly lower perplexity score than a model trained on KGR10, probably because the raw Twitter corpus is too small. Different conclusions on Twitter corpus can be caused by a different method of obtaining tweets by organizers (e.g. filtered by some keywords).

Flair provides also text classifier using a neural network. Word embeddings are fed to a convolutional or recurrent neural network (used in this research).

Two approaches were taken to balance the training data. The first one is oversampling and the second is the usage of weights of classes.

3.2. FastText

FastText uses static word embeddings. Two fastText word embeddings were used:

— trained on KGR10 (Kocoń 2018, Kocoń and Gawor 2018)

— trained on NKJP for 100 epochs.

FastText provides also text classifier which is a linear classifier on averaged word embeddings.

4. Evaluation

The first subtask is evaluated by organizers using F1-score for harmful class and accuracy. The second subtask is evaluated using micro-average F1-score (microF) and macro-average F1-score (macroF). The macro-average F1-score is calculated as harmonic mean of macro-average Precision and Recall. In this paper macro-average F1 is calculated as the average of F1-scores of each class.

5. Experiments and Results

Flair classifier was trained using KGR10 language model with learning rate 0.1 and annealing factor 0.5. The model has a hidden state of size 128, embeddings are projected to 32, word dropout 0.2 and bidirectional LSTM. The training was stopped after 300 epochs or if the score was not improved by 5 epochs on validation data. The fastText classifier was trained using default parameters for 5 epochs.

A baseline for Subtask 1 classifies all data as harmful (class 1) and baseline for Subtask 2 labels all data as non-harmful (class 0).

Table 3 shows scores on training data using 5-fold stratified cross-validation. The folds preserve the percentage of samples for each class. The best result was obtained using fastText classifier trained on NKJP. For Flair oversampling was the best method for class imbalance. For Subtask 2, the baseline achieved a very high score.

Table 3: Results of fastText and Flair classifiers using 5-fold stratified cross-validation on training data

Corpus	**Subtask 1**		**Subtask 2**	
	F1	**Accuracy**	**Micro-average F1**	**Macro-average F1**
fastText NKJP	50.98	93.78	92.58	53.18
fastText KGR10	40.45	92.09	90.79	54.19
Flair	41.87	91.25	90.69	49.95
Flair with weights	40.20	90.95	91.14	40.04
Flair with oversampling	43.54	91.76	91.29	53.00
Baseline	15.63	8.48	91.53	31.86

Table 4 presents results on test data compared with the best systems in PolEval contest. Flair and fastText classifiers achieve similar scores. Pretrained fastText embeddings have influence on F1 score in Subtask 1, but they do not affect micro-average F1 in Subtask 2.

As a final step to Subtask 1, optimization of output class probability with F1 as the objective was performed. 20% of training data was used as validation data for the optimization and the rest was used to train the fastText classifier. The procedure was repeated 10 times and majority voting was used to generate final scores. This result was sent to Subtask 1 named

Table 4: Results of fastText and Flair classifiers compared with baseline and the best systems in PolEval contest. Bolded results were submitted to PolEval.

Corpus	Subtask 1		Subtask 2	
	F1	Accuracy	Micro-average F1	Macro-average F1
fastText	15.89	87.30	86.70	32.19
fastText NKJP	31.64	87.90	**86.80**	**44.04**
fastText KGR10	33.17	86.70	85.10	39.99
Flair	32.10	87.32	86.56	42.06
Flair with weights	34.03	87.26	86.22	33.00
Flair with oversampling	32.48	87.22	86.46	41.79
fastText NKJP optimized	**41.35**	**87.80**	–	–
Baseline	23.63	13.40	86.60	30.94
Best PolEval system	58.58	90.10	87.60	–

`fasttext`. The result obtained using fastText model trained on NKJP was sent to Subtask 2 named `fasttext`.

6. Conclusions

Both approaches did not achieve comparable results with the best systems in the PolEval contest. The cyberbullying detection problem is very complex as the results for Subtask 2 shows in comparison to the simple baseline.

Future works can focus on transfer learning from similar tasks, e.g. sentiment analysis. The larger raw dataset of tweets would be helpful to train language model. For English, Godin et al. (2015) shared word embeddings trained on 400 million tweets. Automatic labeling can be used to obtain more training data, e.g. by matching tweets with vulgarisms in the vocative case. Training data can be augmented by machine translation into another language and back to Polish.

Acknowledgments

This research was supported in part by PLGrid Infrastructure.

References

Akbik A., Blythe D. and Vollgraf R. (2018). *Contextual String Embeddings for Sequence Labeling*. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, pp. 1638–1649.

Bojanowski P., Grave E., Joulin A. and Mikolov T. (2017). *Enriching Word Vectors with Subword Information*. „Transactions of the Association for Computational Linguistics", 5, p. 135–146.

Godin F., Vandersmissen B., De Neve W. and Van de Walle R. (2015). *Multimedia Lab @ ACL WNUT NER Shared Task: Named Entity Recognition for Twitter Microposts using Distributed Word Representations*. In *Proceedings of the Workshop on Noisy User-generated Text*, pp. 146–153.

Joulin A., Grave E., Bojanowski P. and Mikolov T. (2017). *Bag of Tricks for Efficient Text Classification*. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 427–431. Association for Computational Linguistics.

Kocoń J. and Gawor M. (2018). *Evaluating KGR10 Polish Word Embeddings in the Recognition of Temporal Expressions Using BiLSTM-CRF*. „Schedae Informaticae", 2018(27).

Kocoń J. (2018). *KGR10 FastText polish word embeddings*. CLARIN-PL digital repository.

Pęzik P. (2012). *Wyszukiwarka PELCRA dla danych NKJP*. In Przepiórkowski A., Bańko M., Górski R. and Lewandowska-Tomaszczyk B. (eds.), *Narodowy Korpus Języka Polskiego*, pp. 253–279. Wydawnictwo Naukowe PWN.

Ptaszyński M., Leliwa G., Piech M. and Smywiński-Pohl A. (2018). *Cyberbullying Detection – Technical Report 2/2018, Department of Computer Science AGH, University of Science and Technology*. „CoRR", abs/1808.00926.