Results of the PolEval 2017 Competition: Sentiment Analysis Shared Task

Aleksander Wawer and Maciej Ogrodniczuk

Institute of Computer Science, Polish Academy of Sciences
Jana Kazimierza 5, 01-248 Warszawa, Poland
{aleksander.wawer, maciej.ogrodniczuk}@ipipan.waw.pl

Abstract

PolEval is a new SemEval-inspired evaluation campaign for natural language processing tools for Polish. Submitted tools compete against one another within certain tasks selected by organisers, using available data and are evaluated according to pre-established procedures. PolEval 2017 — the first edition of this competition — has included two shared tasks in the area of Part of Speech Tagging and Sentiment Analysis and has gathered 16 submissions from 9 distinct teams. The paper presents the motivation for organizing PolEval, description of the sentiment analysis task, data and measures used for evaluation of systems and their detailed results.

1. Introduction

The abundance of publicly available natural language processing tools and resources for Polish is a fact; the current size of the Computational Linguistics in Poland LRT list (`http://clip.ipipan.waw.pl/LRT`) already exceeded 200 entries and every year brings new improvements, both in terms of analyzed linguistic layers and, presumably, the quality of automatic annotation. Still, for some linguistic tasks there is no clear measure of the quality of the tools neither strict general evaluation metrics which could be used to compare algorithms and methods.

These observations motivated PolEval – the contest for evaluating tools for processing Polish inspired by SemEval (Semantic Evaluation campaign[1], already repeated in several local settings such as GermEval[2]). Such model allows language processing tools to compete against some baseline and one another to extend the current state of the art and provide a forum for the researchers to solve challenging computational linguistic problems. What is equally important, the comparison of systems requires creating high quality annotated datasets and defining evaluation metrics.

For the first edition of PolEval we decided to focus on two tasks: POS tagging and sentiment analysis for Polish. This second summary paper describes the sentiment analysis task, summarized in detail in Section 3. Even though the scope of the competition was Polish, we aimed to engage both local and international participants. PolEval 2017 was advertised among the NLP community in Poland and worldwide by distributing calls for papers to widely recognized discussion groups and mailing lists. Training and test data (see Section 4.) has been released and evaluation measures (see Section 5.) made available two months prior to systems submission date. 16 submissions from 9 distinct teams have been gathered (see Section 6.) and system results have been made announced at PolEval site (`http://poleval.pl`, see also Section 7.). What we are particularly proud of, all winners managed to surpass the state-of-the-art which seems to prove usefulness of our efforts.

[1] See e.g. `http://alt.qcri.org/semeval2017/`.

[2] `https://sites.google.com/view/germeval2017-absa/`.

2. Previous Work

Existing work on sentiment analysis in the Polish language included among other areas, dictionary-based sentiment recognition, aspect-based sentiment analysis. However, according to our best knowledge, there was no previously published machine- or deep-learning tool that used dependency parsing for phrase-level sentiment predictions in the Polish language. The PolEval competition provided the opportunity to design, train and compare such algorithms on the Polish language data sets. For the English language, multiple tools and approaches emerged after publishing the Stanford Sentiment Treebank (Socher et al., 2013). This reference data set contains 9645 sentences from movie review domain. It has been widely used for evaluating multiple deep learning approaches, such as simple recursive neural networks and recently more complex Tree LSTMs (Tai et al., 2015).

3. Task Description

Sentiment analysis is a vital research area, approached at different levels: phrase-level (either in the context of opinion targets/aspects or phrases defined as syntactic subtrees), sentence-level (related to the task of tweet-level analysis).

The aim of this task is to operate on fine-grained levels of words and phrases, investigating how word combinations, captured by dependency syntactic structures, contribute to sentiment formation. The growing stream of works in the English language, backed by emerging deep learning methods, was facilitated by the release of Stanford Sentiment Treebank. Our goal was similar: to promote research on this topic in the context of the Polish language, provide reference data sets to work and motivation for potentially new methods.

Task definition Given a set of syntactic dependency trees, the goal of the task is to provide the correct sentiment for each sub-tree (phrase). Phrases correspond to sub-trees of dependency parse tree. The annotations assign sentiment values to whole phrases (and in some cases, sentences), regardless of their type.

Methods applied to this task often include deep learning. Typically, applications compute sentiment recursively,

starting from leaves and smaller phrases, then expanding to larger phrases and taking into account sentiment values already computed for their nested sub-phrases. This could be equivalent to recursively folding the tree in a bottom-up fashion. Sentence-level sentiment is then the value of your predictive model after folding the whole sentence.

Datasets such as this one include Stanford Sentiment Treebank.

4. Evaluation Data

The evaluation dataset consisted of 350 sentences, manually selected in the Polish Internet according to multiple criteria:

- Sentence complexity. We preferred non-trivial syntax, such as complex clauses or reversed sentiment (e.g. negations) and non-easy phrase-level semantics.
- Topic distribution. We split the sentences into three similar-sized groups: open-domain sentences, perfume review opinionated sentences, clothes-related review opinionated sentences.

The topic structure of the evaluation dataset corresponds to training data, but maintains class balance.

Each sentence in the evaluation dataset was parsed using the Polish dependency parser models available from http://zil.ipipan.waw.pl/PolishDependencyParser. For each sentence, its overall sentiment (neutral, positive and negative), as well as sentiments of each sub-phrase (sub-tree) and each leaf word have been assigned by a linguist. The number of annotated sub-phrases was 2406.

Sentiment annotations for each token corresponded to the overall sentiment of the whole phrase under it and inclusive. Specifically:

- For every leaf token or word, its sentiment corresponds to this word or token's sentiment.
- For every non-leaf token or word (node that has non-empty set of children) sentiment field describes the sentiment of the whole phrase, formed by sub-tree starting at this token (that includes this token and all tokens below it)

The final test data set has not been publicly released before.

5. Evaluation Measures and Baselines

We focused on keeping the evaluation procedures as similar as possible to the approaches used previously by the authors of sentiment analysis systems to report their work. This (apart from different training and testing data) makes the results comparable with previous work and helps to establish long-term best practices in this field.

Given the files with results of dependency syntax analysis for each sentence, participants were to provide sentiment labelings. For leaves — sentiment values of specific words. For non-leaf tokens — sentiment values that reflect the sentiment of the overall phrase, formed by sub-tree starting at this token (that includes this token and all tokens below it).

We compared sentiment labelings obtained from participants with our test set of gold (reference) annotation. We computed micro accuracy between participating systems and gold annotations. Micro accuracy is a global sum of correctly labelled sentiment scores, over cases belonging for each class.

We counted sentiment scores for each leaf and each phrase. We assigned the same weights (1) to all sentiment scores, regardless whether they represented sentiments of leaves, phrases or sentences. We did not distinguish between types of errors.

6. Submitted Systems

Tree-LSTM-NR (Ryciak, 2017) Tree-structured Long Short Term Memory Network (Tree-LSTM) is a kind of neural network designed for tree-structured data which is the generalization of recurrent neural networks (RNN). It works on the principle of propagating information along branches of a tree. The algorithm begins in leaves and moves upwards aggregating information in each node from its subtrees. In the end, network reaches root of the tree and gives an output — prediction for considered tree-structured observation. In particular, Tree-LSTM is the generalization of LSTM network which is widely considered as the best representative of RNNs. The power of LSTM comes from mechanism that supports information propagation through long sequences and this makes Tree-LSTM working efficiently too.

All three sets of predictions were obtained using the TreeLSTM. They differ only in the learning method:

- predictions 1 and 2: During training phase, the neural network model (epoch number) was selected as the best performing one on validation data. The selection of sentences into train and validation sets in predictions 1 and 2 was a bit different in terms of sizes and selection methods.
- predictions 3: The training phase length was determined using early stopping condition set to 5 epochs, measured on test data. Then, we re-trained this model on the combined train and validation sets for 15 more epochs.

As for the training data, the dataset consisted of product reviews of two types (perfumes and clothing) with dependency parse information and sentiment annotations for each phrase (including single tokens and whole sentences) — this data has been provided by the organizers. Additionally, 300-dimensional word2vec vectors trained on combination of Wikipedia and the National Corpus of Polish were used (for words representation).

Tree-LSTM-ML The model was created using architecture of recurrent deep neural network in tree topology. For training data we used syntactic dependency trees with no dependency labels.

The model was built as follows:

- leaves of NN are embeddings of leaves in dependency tree,

- subtrees of NN are embeddings of subtrees in dependency tree and are created by combining embedding of root of dependency subtree one-by-one with its children (from left to right) using LSTM equations.
- embedding of each subtree was projected on its sentiment label.

Alan Turing climbs a tree (Żak and Korbak, 2017) The system used for producing our solution consisted of a Child-Sum Tree-LSTM deep neural network (as described by (Tai et al., 2015)), fine-tuned for working with morphologically rich languages. Fine-tuning included applying a custom regularization technique (zoneout, described by (Krueger et al., 2016), and further adapted for Tree-LSTMs) as well as using pre-trained word embeddings enhanced with sub-word information (fastText, (Bojanowski et al., 2016)). Our solution was implemented in PyTorch.

Pawsemble (Rychlikowski and Zapotoczny, 2017) First, we used our POS tagger to tag and lemmatize the data. After that we applied the simple ensemble learning, joining the two following methods:

1. Linear regression working on the whole phrases represented as bag of lemmas (with sparse 0/1 encoding, only lemmas with frequency greater than 1 were considered).
2. Simple counting base tagger implemented as a sequence of NLTK tagger (DefaultTagger returning 0, Unigram-, Bigram-, and TrigramTagger). This tagger took as an input sequence of lemmas and returns the sequence of 'tags' (0,-1,+1). It does not consider any other information from data.

Having the result from the above method we integrate them using the simple rule: if the first model says 'Neutral' we give the result of the second model, otherwise the result of the first model is returned.

Paschał (Zapotoczny and Rychlikowski, 2017) In this approach we used our POS tagger to tag and lemmatize the available data. Then, in training data we find lemmas that start a sentiment path (a sentiment path is a path in dependency tree in which all the nodes have ANY sentiment). We also find lemmas that are on sentiment path and divide them into to categories: passing lemmas and non-passing lemmas. In second category we have words that end sentiment path, all other words that are in any sentiment path go to the first category. During evaluation we first mark all starting lemmas. Then we grow sentiment paths by applying passing lemmas.

For training we used data given by the competition organizers. During testing, we used 10-fold cross validation.

7. Evaluation Results

To evaluate the performance of the submitted systems, we have calculated their accuracy against the provided test data and compared them with existing systems, proposed to date.

The results for Task 2 are presented in Table 1. Most of participating systems were based on deep learning and neural networks. The most popular architecture, used by teams (Ryciak, 2017) and (Żak and Korbak, 2017) was based on variants on Tree LSTM adjusted for dependency parse trees (usually, by averaging over hidden anc cell states of all node's children). Competing systems were heavily tuned to obtain the highest possible prediction quality using such regularization techniques as dropout, zone-out or L2. The winning system was based on Tree LSTM using word2vec embeddings generated for the Polish language.

8. Summary

PolEval 2017 was the first edition of a NLP shared task aimed at encouraging work on tools and resources focusing on Polish language and disseminating it in the form of scientific publications and open-source tools. PolEval 2017 has resulted in providing 2 new language resources (tasks evaluation data) and has attracted 16 submissions from 9 teams in total. The systems submitted for the competition are currently the best-performing state-of-the-art methods in Polish.

Apart from the accuracy of the systems, some of the authors have declared that they focused on the speed and robustness of their methods in the context of Big Data and distributed computing. In the current edition of PolEval we have focused on evaluating accuracy exclusively, but this aspect in future editions of the competition remains to be considered.

Many of the competing and winning systems were based on emerging deep learning techniques and algorithms. The PolEval competition had turned out to be an impulse for the development of this kind of NLP tools and practicing their usage on the Polish language data sets. In many cases it resulted in out-performing older solutions, usually based on machine learning.

Hopefully, future editions of PolEval will include at least the same topics as the current one. However, the list may include other important areas such as syntactic parsing (e.g. dependency) and perhaps cover other syntactic and semantic tasks.

Acknowledgments

We would like to thank all participants who have made PolEval 2017 a very successful event and LTC 2017 (Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics, `http://ltc.amu.edu.pl`) conference organizers for welcoming PolEval 2017 results in a special session.

Finally, we gratefully acknowledge the support of Sages, the sponsor of the prize for the winner of the task.

This work has also been financed as part of the investment in the CLARIN-PL research infrastructure funded by the Polish Ministry of Science and Higher Education.

9. References

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov, 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Table 1: Evaluation of Task 2: micro accuracy of each system.

System	Variant	Accuracy
Tree-LSTM-NR	(predictions 2)	0.795
Tree-LSTM-NR	(predictions 3)	0.795
Tree-LSTM-NR	(predictions 1)	0.779
Pawsemble	(results)	0.770
Paschał	(output-labels)	0.769
Tree-LSTM-ML	(test_file)	0.768
Alan Turing climbs a tree	(fast_emblr)	0.678
Alan Turing climbs a tree	(slow_emblr)	0.671
Alan Turing climbs a tree	(ens)	0.670

Krueger, David, Tegan Maharaj, János Kramár, Mohammad Pezeshki, Nicolas Ballas, Nan Rosemary Ke, Anirudh Goyal, Yoshua Bengio, Hugo Larochelle, Aaron C. Courville, and Chris Pal, 2016. Zoneout: Regularizing rnns by randomly preserving hidden activations. *CoRR*, abs/1606.01305.

Rychlikowski, Paweł and Michał Zapotoczny, 2017. Pawsemble. Unpublished.

Ryciak, Norbert, 2017. Polish language sentiment analysis with tree-structured long short-term memory network. In Zygmunt Vetulani (ed.), *Proceedings of the 8th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*. Poznań, Poland.

Socher, Richard, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts, 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA: Association for Computational Linguistics.

Tai, Kai Sheng, Richard Socher, and Christopher D. Manning, 2015. Improved semantic representations from tree-structured long short-term memory networks. In *ACL 2015*.

Żak, Paulina and Tomasz Korbak, 2017. Fine-tuning tree-LSTM for phrase-level sentiment classification on a Polish dependency treebank. Unpublished.

Zapotoczny, Michał and Paweł Rychlikowski, 2017. Paschał. Unpublished.