

Sentiment Analysis

ML[†]
PeP*

[†]VoiceLab
michal.lew@voicelab.pl

*University of Lodz
piotr.pezik@uni.lodz.pl

Abstract

This paper evaluates thsdfh.

1. Introduction

Sentiment Analysis (SA) is an active area of natural language processing with applications in general opinion mining, customer relations management systems, marketing and social media communication studies. A common objective of SA is to automatically detect the attitudinal value of an utterance or otherwise coherent stretch of text which can be attributed to a particular author. The distribution of such attitudinal values in an utterance is usually known as its *polarity* as it usually ranges from positive to neutral and negative.(Cambria et al., 2013). More recent approaches to SA stress the importance of phrase-level polarity classification, whereby attitudinal values are detected at the level of syntactic phrases and subsequently compositionally combined to obtain the polarity of an entire utterance. Such approaches are particularly important in aspect-based SA, where different aspects of a product or other objects of one’s opinions are important to detect and classify. Also, phrase-level SA is better suited to deal with basic syntactic operations such as negation or modality which may cause a significant shift in the predominant sentiment of an utterance.(Wilson et al., 2009).

2. PolEval 2017

W	Nie	drażni	nawet	po	całym	dniu	.
P	2	0	2	2	6	4	2
S	0	1	0	0	0	0	0

Table 1: An example PolEval dataset sentence with phrase-level sentiment annotation (W – word nodes, P – syntactic parents, S – polarity value).

3. Model

3.1. LSTM

Recurrent Neural Networks has been widely used for many task in natural language processing. They can process the arbitrary long inputs by recurrent application of transition function over hidden state. The most common form of RNN’s transition function is an

affine transformation followed by hyperbolic tangent function

$$h_t = \tanh(W_{x_t} + U_{h_{t-1}} + b) \quad (1)$$

This gives them ability to use the information gathered in previous states when processing text. In theory it makes them able to look at the infinite history while processing sequences of words. Unfortunately they suffer from so called vanishing gradient problem, which means that during training gradient can grow or decay exponentially over long sequences [s (Hochreiter, 1998; Bengio et al., 1994)] The LSTM architecture (Hochreiter and Schmidhuber, 1997) addresses this problem of learning long-term dependencies by introducing a memory cell that is able to preserve state over long periods of time. [citation] There are many different variants of LSTMs, in our work we used following equations: [równania LSTM] where x_t is input at current timestep, i_t is an input gate, f_t a forget gate, o_t an output gate and c stands for LSTM memory. The gates mechanism controls how much information from past state and memory is used at current timestep.

3.2. Tree-LSTM

The problem with classical LSTM’s is that they are linear chain which makes them useless for processing more sophisticated structures than sequences. In particular, many linguistic beings are represented with tree-structure [citation?]. In order to process tree-structured data we need to extend the LSTM architecture. In our work we propose Sequential-Child-Combination Tree-LSTM. In general, tree-structured Recurrent Neural Networks expand basic RNNs by creating a way to plug in more than one previous state to hidden unit at current time step. [obrazek podłączanie 2 poprzednich stanów do nowego] The essence lie in the way in which the children states are combined with parent state.

3.3. Child-Sum Tree-LSTM

Child-Sum Tree-LSTM equations

3.4. Sequential-Child-Combination Tree-LSTM

In our work we combine children with its parent as follows:

4. Evaluation

Number of tags (without <i>ign</i>)	35
Token accuracy	98.2%
Training set	PolEval TR
Test set	PolEval TE
Training time	5m 40s (5 iterations)
Tagging time	1.42s

Table 2: Speed and accuracy evaluation of an AP-based ‘flexeme’ tagger.

5. Conclusions and future work

6. Availability

The source code of the tagger described in this paper, together with the trained models, datasets, tagset translations and other resources such as the NCP Brown clusters are publicly available at https://gitlab.com/piotr.pezik/apt_pl.

7. References

- Cambria, E., B. Schuller, Y. Xia, and C. Havasi, 2013. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*, 28(2):15–21.
- Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann, 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(3):399–433.