Detecting Figurative Word Occurrences Using Recurrent Neural Networks

Agnieszka Mykowiecka
ICS PAS
Jana Kazimierza 5
Warsaw, Poland
agn@ipipan.waw.pl

Aleksander Wawer
ICS PAS
Jana Kazimierza 5
Warsaw, Poland
axw@ipipan.waw.pl

Małgorzata Marciniak
ICS PAS
Jana Kazimierza 5
Warsaw, Poland
mm@ipipan.waw.pl

Abstract

The paper addresses the detection of figurative usage of words in English text. The chosen method was to use neural nets fed by pre-trained word embeddings. The obtained results show that simple solutions, based on word embeddings only, are comparable to complex solutions, using additional information as a result of taggers or a psycholinguistic database. This approach can be easily applied to other languages, even less-studied, for which we only have raw texts available.

1 Introduction

Natural language is a very efficient way of communication. To make the task of learning and remembering language easier, the same linguistic expression can have many different meanings, e.g. *the nearest bank*. What is more, in spite of regular homonymy and polysemy, words or expressions can have a meaning that is different from all literal interpretations. The latter phenomena, called figurative usage, allows for much more creative and rich communication, and makes it more effective, persuasive, and impactful. It is very often used in poetry or literature, but is also quite frequent in everyday language. Although figurative meanings are different from literal ones, there usually exists some linkage between both meanings which make metaphors comprehensive for a hearer/reader. For example, when somebody says *I am a rock* we start to think about being hard and solid. Thus, we can easily understand not just conventional figurative expressions which we already know, but also those that we read or hear for the first time.

The problem which we tried to solve was defined by the organizers of the Figurative Language NAACL Workshop shared task in which we took part as the ZIL-IPIPAN team. In this task, participants were supposed to label, in a given subset of VU Amsterdam Metaphor Corpus (Steen et al., 2010), individual words which were used metaphorically. As people are able to recognize metaphorical usage of a word based on the actual context, we decided to test to what degree it is possible to automatically recognize metaphorical word occurrence using only word embeddings.

2 Related Work

Multiple approaches have been proposed for the problem of detecting metaphors in text. Among many published methods, we only discuss selected ones in this section, especially those based on the Amsterdam metaphor dataset.

In (Beigman Klebanov et al., 2016), the authors apply a logistic regression classifier to test combined lexical and dictionary-based feature spaces.

In (Rai et al., 2016), a conditional random field (CRF) algorithm is proposed. The approach is based on features from the MRC psycholinguistic dictionary (Wilson and Division, 1997) and WordNetAffect database (a subset of WordNet with emotion annotations).

Perhaps the the method described in (Do Dinh and Gurevych, 2016) is the most relevant to our work, where a neural network is used to recognize word-level metaphoricity. As in our approach, word embeddings are used to represent words. However, the structure of the network is different: it is a dense multi-layer network, while we focus on recurrent networks (such as LSTM), in our opinion more suitable for labelling sequential, word-level data. Interestingly, the authors demonstrate the positive influence of part-of-speech (POS) based features, used to augment word embeddings. The best overall model is based on combining word embeddings, POS and selected MRC dictionary data.

Proceedings of the Workshop on Figurative Language Processing, pages 124–127
New Orleans, Louisiana, June 6, 2018.

3 Data

The texts in the VU AMC corpus, used in the shared task, originated from the British National Corpus from four genres: News, Fiction, Academic and Conversation. VU AMC was divided into two parts: train and test. The train set was used to prepare classifiers of metaphorical and literal senses of tokens, while a test set was used for evaluation. The numbers of sentences tokens and metaphors of both parts are given in Table 1.

part	sentences	tokens	metaphors	% of met.
train	8,883	106,986	9,022	8.43
test	4,080	58,359	6,822	11.69

Table 1: The test and train datasets in numbers

The solutions were tested on 22,196 tokens from the test set indicated by the organizers.

4 Neural Net Architecture

In our experiments, we adopted the method described in (Wawer and Mykowiecka, 2017) as a starting point. The authors applied neural networks and word embeddings to predict if a noun-adjective phrase has a literal or metaphorical sense or can have both senses depending on its usage. As the current task concerns labelling all words in a sentence, the obvious choice was to use a sequential model. We tested both GRU and LSTM units in a bidirectional architecture, as the important information may be coded both in left and right word context. The implementation is done in Keras with the Tensorflow backend – the model summary is given in Figure 1. The sequential network has to be of a fixed length, thus the maximum length of the sentence was chosen (to be equal to 110). As word representation, we used 300 element GLoVe vectors trained on Wikipedia 2014 and Gigaword 5 (Pennington et al., 2014)

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 110, 300)          0
_________________________________________________________________
bidirectional_1 (Bidirection (None, 110, 600)          1442400
_________________________________________________________________
dropout_1 (Dropout)          (None, 110, 600)          0
_________________________________________________________________
bidirectional_2 (Bidirection (None, 110, 600)          2162400
_________________________________________________________________
dense_1 (Dense)              (None, 110, 3)            1803
=================================================================
Total params: 3,606,603
Trainable params: 3,606,603
Non-trainable params: 0
```

Figure 1: Basic net architecture

As it might be correct that the information included in word embeddings is not sufficient, we tested the impact of additional information. We extended appropriate word embeddings with more features. Two types of information were considered. First, we added morphological information about part of speech categories. Second, we used information from General Inquirer data.

4.1 Adding part-of-speech data

In our experiments, we tested if enriching data by part-of-speech (POS) had a positive effect on the results. At the beginning, we wanted to extract POS from the xml file of VU AMC available on the shared task page, but it occurred that it contained tokens/parts omitted in the train and test text files, and the tokenization was inconsistent in the text and xml datasets. Because we were not sure of all the changes made to the text data, we tagged the train and test texts with the Stanford tagger (Toutanova et al., 2003) available from `https://nlp.stanford.edu/software/tagger.shtml`, and we applied the bidirectional model. As the tokenization used in the tagger divided strings into finer ones in comparison to VU AMC, we removed redundant tags where it was necessary. For example, in the corpus, there were amounts of money given by one token *£10,000* but the tagger divided them into two tokens: *£* tagged as '#' and *10,000* tagged as 'CD'. As we had to choose one tag we deleted the first one and left the second. There were many similar differences, especially in tokenization of strings containing a digit.

4.2 Adding General Inquirer Data

It has been shown that using information from external dictionaries may be beneficial for training models on the metaphor detection problem. In their baseline paper (Beigman Klebanov et al., 2016) demonstrate the positive influence of features derived from the WordNet dictionary.

For this task, some researchers use not only general purpose dictionaries (such as WordNet) but also more specialized, psychological and psycholinguistic databases of words. For example, the MRC database (Wilson and Division, 1997), a large dictionary listing linguistic and psycholinguistic attributes obtained experimentally, has been applied to metaphor detection in a cross-lingual model transfer scenario (Tsvetkov et al., 2014).

In our experiments, we used another such database: The General Inquirer (Stone et al., 1966). The dictionary (a total of 183 categories assigned to over eleven thousand words that cover a large part of the commonly used English lexicon) contains two sub-parts: the Harvard IV psychosocial dictionary and the Laswell dictionary of values in politics. We conducted our experiments using the Harvard IV part. It contains all three Osgood dimensions (including evaluative dimension, often called sentiment, but also potency and activity), and also many other categories related to pleasure, pain, emotions, various social institutions (sport, politics, religion) and social cognition, cognitive orientation, and emotional states. A more comprehensive description and listing of the categories can be found at http://www.wjh.harvard.edu/~inquirer/homecat.htm. The dictionary is only available for English. Its translation would be a complex and challenging task. This might involve validation against many perspectives, both theoretical and empirical, as many groups of researchers contributed their parts of the dictionary over decades. For example, Osgood labels come from factor analysis of a large survey, Laswell dictionary labels are grounded in studies of totalitarian regimes.

We tested for the presence of each input word in the General Inquirer dictionary and created binary input vectors for neural network models, with a '1' indicating that the word belongs to a given category and a '0' otherwise.

5 Results

The main neural net architecture was chosen based on the experience with solving other tasks and data sets (see (Mykowiecka et al., 2018); recognition of figurative/metaphorical senses of Polish phrases in sentences, recognition of temporal relations — work in progress), but still some decisions had to be made as to the number of layers, the number of epoch, and the degree of the dropout. To select the best configuration we planned to perform 10-cross validation on the training data. As our experiments with LSTM networks were time consuming, we eventually decided not to perform them on all 10 folds but on their subset. The exact number of folds are given in Table 2. The results of these preliminary experiments are given in Table 2. The results show that the LSTM units are better than GRU. The larger number of layers (3 instead of 2) helped slightly for the LSTM network and worsened the results of the GRU network. For the GRU architecture, the 15 epochs are better than 10 or 20; for LSTM, 10 epochs turned out to be the best choice of those three values. Adding information on POS tags helped in the case of the GRU network and had very little influence on the results of the LSTM architecture. The same slight, positive, influence was observed after adding either 20 or 50 features from the General Inquirer to the input of the LSTM network.

type	folds	acc.	P	R	F1
GRU					
	2 layers, 15 epochs, dropout 0.4				
	10	-	0.71	0.62	0.66
	2 layers, 20 epochs, dropout 0.4				
	2	-	0.71	0.60	0.65
	3 layers, 15 epochs, dropout 0.4				
	10	-	0.70	0.61	0.65
	3 layers, 10 epochs, dropout 0.4 + POS tags				
	1	0.982	0.68	0.70	0.69
LSTM					
	2 layers, 10 epochs, dropout 0.4				
	10	0.985	0.74	0.72	0.73
	2 layers, 15 epochs, dropout 0.4				
	10	-	0.71	0.62	0.66
	3 layers, 10 epochs, dropout 0.4				
	4	0.984	0.73	0.71	0.72
	3 layers, 20 epochs, dropout 0.4				
	2	0.982	0.73	0.62	0.67
	2 layers, 10 epochs, dropout 0.4 + POS tags				
	5	0.985	0.75	0.72	0.74
	2 layers, 20 epochs, dropout 0.3 + POS tags				
	10	0.985	0.76	0.71	0.74
	3 layers, 10 epochs, dropout 0.4 + POS tags				
	4	0.984	0.74	0.71	0.73
	2 layers, 10 epochs, dropout 0.4 + GI20				
	5	0.985	0.74	0.72	0.73
	2 layers, 5 epochs, dropout 0.4 + GI50				
	10	0.985	0.76	0.71	0.73
	2 layers, 10 epochs, dropout 0.4 + GI50				
	10	0.984	0.75	0.72	0.73
	2 layers, 10 epochs, dropout 0.3 + POS tags + GI50				
	10	0.985	0.76	0.70	0.73

Table 2: Results of partial 10-fold cross validation on train data set, all-pos task; folds – number of folds processed. GI stands here for the features taken from the General Inquirer. The number indicates how many (beginning) features were taken. POS indicates adding the encoded part of the speech tag.

We applied the models trained on the entire training data on the test data and observed slightly different results (see Table 3). However, the LSTM architecture still turned out to be more effective, generally, and the obtained results were lower that those from the cross-validation schema. The best results (0.58 for all words and 0.62 for

type	lrs	dpt	ep.	add-inf	F1:all	F1:v
LSTM	2	.4	10	-	0.583	0.619
LSTM	2	.4	15	-	0.574	0.602
LSTM	3	.4	10	GI20	0.545	0.563
LSTM	3	.5	7		0.541	0.553
LSTM	3	.4	10	-	0.536	0.544
LSTM	2	.4	7	GI_POS	0.518	0.543
GRU	3	.5	15	-	0.514	0.561
GRU	2	.4	20	-	0.506	0.546
GRU	2	.5	15	-	0.485	0.524
LSTM	3	.4	10	POS	0.475	0.558
LSTM	1	.4	5	-	0.447	0.450
GRU	3	.5	20	-	0.425	0.452
LSTM	1	.4	5	GI50	0.350	0.338

Table 3: Results on the test set ordered by the F1 value (for metaphors only) for the all-pos task. Models differ in type of unit network, number of layers, size of dropout, number of epochs and the type of additional information included apart from embeddings. GI stands here for the features taken from the General Inquirer. The number indicates how many beginning features were taken. POS indicates adding the encoded part of the speech tag.

verbs) were obtained using the model which was not the best one in the cross-validation schema but, nevertheless, it obtained an F-value equal to 0.72 on all the words. In the case of the test data, adding POS names and features from the General Inquirer worsened the results.

6 Conclusions

Recurrent sequential neural networks turned out to be capable of recognizing metaphorical usage of words better than many other already tested approaches. The exact result achieved – F1 equal to 0.73 for the metaphorical words and to 0.58 for the test data in the cross-validation schema – shows that the scores are not very stable and, probably, the optimal net architecture and settings were not already found. An improvement in the results after adding General Inquirer data, at least for some configurations, shows that the enrichment of the vector representation by additional features might be effective and that this idea needs further study.

Acknowledgments

This work was supported by the Polish National Science Centre project 2014/15/B/ST6/05186 and partially as a part of the investment in the CLARIN-PL research infrastructure funded by the Polish Ministry of Science and Higher Education.

References

Beata Beigman Klebanov, Chee Wee Leong, E Dario Gutierrez, Ekaterina Shutova, and Michael Flor. 2016. Semantic classifications for detection of verb metaphors. In *Proceedings of ACL*, pages 101–106. ACL.

Erik-Lân Do Dinh and Iryna Gurevych. 2016. Token-level metaphor detection using neural networks. In *Proceedings of the Fourth Workshop on Metaphor in NLP*, pages 28–33, San Diego, California. ACL.

Agnieszka Mykowiecka, Małgorzata Marciniak, and Aleksander Wawer. 2018. Literal, metphorical or both? detecting metaphoricity in isolated adjective-noun phrases. In *Proceedings of Workshop on Figurative Language Processing*. ACL.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Sunny Rai, Shampa Chakraverty, and Devendra K. Tayal. 2016. Supervised metaphor detection using conditional random fields. In *Proceedings of the Fourth Workshop on Metaphor in NLP*, pages 18–27, San Diego, California. ACL.

Gerard J. Steen, Aletta G. Dorst, J. Berenike Herrmann, Anna Kaal, Tina Krennmayr, and Trijntje Pasma. 2010. *A method for linguistic metaphor identification. From MIP to MIPVU.* Number 14 in Converging Evidence in Language and Communication Research. John Benjamins.

Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. The MIT Press.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of NAACL HLT*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. ACL.

Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of ACL (1)*, pages 248–258. ACL.

Aleksander Wawer and Agnieszka Mykowiecka. 2017. Detecting metaphorical phrases in the Polish language. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 772–777, Varna, Bulgaria. INCOMA Ltd.

Michael Wilson and Informatics Division. 1997. Mrc psycholinguistic database: Machine usable dictionary, version 2.00. 20.