Multi-Module Recurrent Neural Networks with Transfer Learning. A Submission for the Metaphor Detection Shared Task

Filip Skurniak
Samsung R&D Poland
pl. Europejski 1
00-844 Warszawa, Poland

Maria Janicka
Samsung R&D Poland
pl. Europejski 1
00-844 Warszawa, Poland

Aleksander Wawer
Samsung R&D Poland
pl. Europejski 1
00-844 Warszawa, Poland

{f.skurniak,m.janicka,a.wawer}@samsung.com

Abstract

This paper describes multiple solutions designed and tested for the problem of word-level metaphor detection. The proposed systems are all based on variants of recurrent neural network architectures. Specifically, we explore multiple sources of information: pretrained word embeddings (Glove), a dictionary of language concreteness and a transfer learning scenario based on the states of an encoder network from neural network machine translation system. One of the architectures is based on combining all three systems: (1) Neural CRF (Conditional Random Fields), trained directly on the metaphor data set; (2) Neural Machine Translation encoder of a transfer learning scenario; (3) a neural network used to predict final labels, trained directly on the metaphor data set. Our results vary between test sets: Neural CRF standalone is the best one on submission data, while combined system scores the highest on a test subset randomly selected from training data.

1 Introduction

1.1 Shared Task

This paper is focused on the problem of automated metaphoricity classification of verbs. It describes a system aimed at the Shared Task https://competitions.codalab.org/competitions/17805 on metaphoricity classification co-organized with the Workshop on Figurative Language Processing.

The task is based on VUA Metaphor corpus (Steen et al., 2010). The data set, as its authors claim, is the largest available corpus hand-annotated for all metaphorical language use, regardless of lexical field or source domain. The method of metaphor labeling is consistent with systematic and explicit metaphor identification protocol MIPVU. The corpus consists of altogether 117 texts covering four genres (academic, conversation, fiction, news).

Our submissions and results are for the all POS (part-of-speech) part of the task.

2 Existing Work

2.1 Predicting Metaphoricity

The VUA Metaphor Corpus has been previously used to automatically predict the metaphoricity of verbs. In the baseline paper (Klebanov et al., 2016) authors explore multiple feature spaces, based on VerbNet and WordNet databases, clustering distributional similarity data of verbs. Tested classifiers included Logistic Regression, Random Forest and Linear SVM. The best of reported F1 scores averaged over four document types in the VUA corpus reach 0.60 for a feature space combined of lemma unigrams and WordNet data.

In another study (Rai et al., 2016) authors use a Conditional Random Field algorithm and a feature space of MRC and WordNetAffect dictionaries.

In Do Dinh and Gurevych (2016) a neural network based on word embeddings is used to detect metaphorical words. The network is a multi-layer one, but not sequential as in our approach.

In a similar manner, (Sun and Xie, 2017) use four sequential recurrent neural networks (bi-LSTM) to predict metaphors. The first three models use a sub-sequence as the input to BiLSTM network, each with a special kind of sub-sequence extracted from the input sentence. The last model is an ensemble model which aggregates the outputs from the first three models.

2.2 Transfer Learning

The idea of transfer learning has not been widely explored in the context of predicting the metaphoricity, especially in the context of verbs. We do not consider the method described in Bizzoni et al. (2017) to be fully transfer learning.

In our understanding, the term transfer learning refers not only to finding representations of words

Proceedings of the Workshop on Figurative Language Processing, pages 128–132
New Orleans, Louisiana, June 6, 2018. ©2018 Association for Computational Linguistics

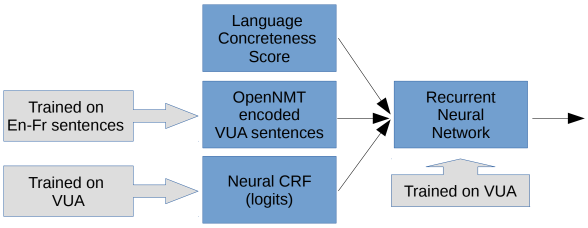
Language Concreteness Score
Trained on En-Fr sentences
OpenNMT encoded VUA sentences
Recurrent Neural Network
Trained on VUA
Neural CRF (logits)
Trained on VUA

Figure 1: System design

in some vector space, but also to training full models that solve some non-trivial sequential problem, in order to apply them later to another one. Our approach is similar to Conneau et al. (2017) where authors investigate transfer learning to find universal sentence representation. The concept is to use datasets originally compiled for different applications, such as question answering, textual entailment or sentiment analysis, to finally apply them to some other task (in Conneau et al. (2017), to find sentence representation).

3 System Design

We test multiple systems and components on the task of word-level metaphor recognition. The architecture is based on multiple components that constitute input space for a recurrent neural network, which produces output labels. It combines the following elements: (1) Neural CRF (Conditional Random Fields), trained directly on the metaphor data set; (2) Neural Machine Translation encoder, used in the transfer learning scenario; (3) a neural network to predict final labels, trained on the metaphor data set. Figure 1 illustrates the system. Elements (1) – the neural CRF and (3) – the recurrent network can be used to predict the output labels and we test them both in subsequent sections.

3.1 Neural CRF

We used a sequence tagging model (Ma and Hovy, 2016) to generate scores (logits) for each tag. We used those logits for directly predicting the output labels as well as for input features into another recurrent network. The model is based on both word representation and contextual word representation. The former uses pre-trained word embeddings (GloVe (Pennington et al., 2014) trained on Wikipedia 2014 and Gigaword-5 corpus) as well as features on the character level extracted using bidirectional LSTM (Hochreiter and Schmidhuber, 1997). The latter is based on bidirectional LSTM on the word level, which captures information about the context. In the decoding phase, the vector of scores corresponding to each tag is generated with a fully connected neural network. Finally, predictions are made with linear-chain CRF which, in contrast to a simple softmax function, make use of the neighboring tagging decisions.

We fed the presented model with training data from the VUAMC corpus. The model has been used in two settings: standalone, to directly predict the output labels, and in another mode, where we used the extracted logits (the output of a fully connected neural network on an encoded state of bidirectional LSTM on words level) as an input for another recurrent neural network, as illustrated in Figure 1.

3.2 Concreteness Score

We used the concreteness score from Brysbaert et al. (2014) database, which provides ratings for nearly 40,000 words. For each word, its mean concreteness rating, ranging from 1 to 5, was computed based on at least 25 observations. In the task instructions, concreteness was defined as a feature of words related to things and actions which can be experienced directly through senses. In addition, the task designers put stress on all 5 modalities, providing examples of concrete words connected with different senses.

In our data set we found concreteness scores for nearly 66% of words. For those that could not be found in Brysbaert et al. (2014) database we assumed a mid value of 2.5 as a neutral score. We later normalized these values.

MIPVU (Metaphor Identification Procedure VU University Amsterdam) (Steen et al., 2010) is based on investigating if there is a more basic, concrete, body-related, precise or historically older meaning of a given word compared to its contextual meaning. The concreteness score may indicate if the contextual meaning of a token is also its basic meaning.

3.3 OpenNMT encoded VUA Sentences

OpenNMT (Klein et al., 2017) `http://opennmt.net` is an initiative for neural machine translation and neural sequence modeling. It offers a set of tools dedicated for machine translation, which enable end-to-end translation process are offered.

In our solution the OpenNMT implementation is used in a transfer learning fashion: a model

	Measures			Features				
	P	R	F1	Conc.	Logits	Encoder states	GloVe 100	GloVe 300
bi-GRU 3 layers	0.57	0.67	0.62	x	x	x	x	
bi-GRU 3 layers	0.61	0.51	0.55	x	x	x		
bi-GRU 2 layers	0.61	0.63	0.62	x	x	x	x	
bi-GRU 2 layers	0.59	0.5	0.54		x	x	x	
bi-GRU 2 layers	0.66	0.52	0.58	x	x			
bi-GRU 2 layers	0.57	0.58	0.57				x	
neural CRF	0.58	0.57	0.57					x

Table 1: Best training phase scores (all POS).

trained for machine translation is used to generate a representation of an input sentence. Then, instead of translating the sentence into another output language, we use the intermediate representation for metaphor recognition.

Thus, the overall procedure was to (1) train the translation model; (2) translate Metaphor Shared Task sentences and capture the hidden states of a machine translation encoder for each sentence and (3) extract the hidden vector for every word.

1. Training translation model

   With the aim to maximize usability of the model and consequently, quality of the extracted encoder states, we decided not to use pre-trained models available in the web but rather to use an open source dataset of parallel sentences instead. The corpora are provided by Tiedemann (2012) and are commonly used in the machine translation tasks.

   The translating model is trained on one million English sentences with their French translations.

2. Translation and hidden states

   The translating model consists of a encoder-decoder approach. The model used in the solution is built with simple unidirectional LSTM. The hidden states of the LSTM were captured during the translation process. Typically, the outputs of the encoder play the role of an intermediate layer in the translation process. The encoded states capture the meaning of a sentence.

3. Word vectors extraction

   Extracting word vectors is the last step of the process. Finally, each word is represented by a 500-dimensional vector.

3.4 Bidirectional GRU

To predict metaphors in a given text we used bidirectional Gated Recurrent Units (GRU). Previously described features - concreteness score, logits from neural CRF and OpenNMT hidden states - as well as pre-trained words embeddings (GloVe) served as an input to our neural network.

4 Results

All reported results were obtained for all part-of-speech data.

4.1 Training Phase

Initially, we evaluated different versions of our model on the provided training set - randomly shuffled and divided into three subsets (15% test / 15% - validation / 70% - training). The results on this test set (not the Shared Task official test set) are presented in Table 1.

We tested the models with a different number of layers and sets of features. Models with all features showed the best performance. Omitting any of them led to a considerable decrease in F1 score. We also tried class weighting which slightly increased the performance. Finally, we tested neural

	Measures			Features					
	P	R	F1	Conc.	Logits	Encoder states	GloVe 300	GloVe 100	class weighting
bi-GRU 3 layers	0.722	0.312	0.435	x	x	x			
bi-GRU 3 layers	0.705	0.343	0.461	x	x	x			x
bi-GRU 3 layers	0.675	0.371	0.479	x	x	x		x	x
bi-GRU 2 layers	0.655	0.237	0.348					x	
bi-GRU 2 layers	0.638	0.407	0.497	x	x	x		x	x
bi-GRU 2 layers	0.621	0.362	0.457					x	
neural CRF	0.547	0.575	0.561				x		

Table 2: Best submission scores (all POS).

CRF and bidirectional GRU with GloVe embeddings. Those more basic models served as a point of reference.

The best score was generated by a bidirectional GRU with all the features. A difference in layers number did not show any significant change in performance.

Batch sizes for all models were set to 64 or 128 during experiments. Models were trained using Adam optimizer and a binary cross-entropy loss function.

The network named 'bi-GRU 2 layers' in Table 1 contained two bi-directional LSTM layers. Dropouts were applied after each layer with rates in range from 0,5 to 0,6. Bi-directional layers were followed by two dense layers of size 500 with dropouts (rate 0,5) placed after each of them. The last layer of this network was a sigmoid one. All GRU layers had 'tanh' activation functions, dense layers 'relu' activation functions.

The network named 'bi-GRU 3 layers' in Table 1 contained three bi-directional LSTM layers followed by a sigmoid layer. Dropouts were applied after each bidirectional layer, with rates in range from 0,5 to 0,6 as before.

4.2 Submission Phase

Table 2 shows our submission scores obtained by the best performing models chosen in the previous step. We tested them on the all part-of-speech task.

Interestingly, scores from submission differ significantly from those observed in the training phase. Here, the Neural CRF model applied standalone came out as the best solution. Three layer bidirectional GRU generated a better F1 score than two layers model. However, both models gained much lower scores than noted in the training phase.

This discrepancy can be possibly explained by different character of our test set (random sub-part of the training data set), compared to the official test set in the shared task.

5 Conclusions

In this paper we have discussed solutions for metaphor detection built for Metaphor Detection Shared Task. We described different features and architecture combinations along with their scores, measured on a test set randomly sampled from training data and on official submission procedure.

Due to discrepancies between scores obtained in from the training set and scores obtained in submission, it is not easy to draw straightforward conclusions.

When tested on a subset of training data, our results indicate that all proposed features: those captured in OpenNMT encoder states, concreteness ratings and tag scores from neural CRF, all had an impact on the performance of our system, which resulted in a better F1 score than simple models using GloVe. These results seem to go along the lines of results reported in Do Dinh and Gurevych (2016).

Submission results, as measured on the official test set of the Shared task, provide an entirely different picture. They also show the advantage of bidirectional GRU including all features over one trained on GloVe only. Yet, it is neural CRF standalone, which included only pre-trained embeddings, that outperformed other more complex models.

References

Yuri Bizzoni, Stergios Chatzikyriakidis, and Mehdi Ghanimifard. 2017. ”deep” learning : Detecting metaphoricity in adjective-noun pairs. In *Proceedings of the Workshop on Stylistic Variation*, pages 43–52, Copenhagen, Denmark. Association for Computational Linguistics.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior Research Methods*, 46(3):904–911.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *CoRR*, abs/1705.02364.

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. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Beata Beigman Klebanov, Chee Wee Leong, E. Dario Gutiérrez, Ekaterina Shutova, and Michael Flor. 2016. Semantic classifications for detection of verb metaphors. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, 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. Association for Computational Linguistics.

Gerard J. Steen, Aletta G. Dorst, J. Berenike Herrmann, Anna Kaal, Tina Krennmayr, and Trijntje Pasma. 2010. *A Method for Linguistic Metaphor Identification*. John Benjamins Publishing.

Shichao Sun and Zhipeng Xie. 2017. Bilstm-based models for metaphor detection. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 431–442. Springer.

Jrg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).