Evaluating a Bi-LSTM Model for Metaphor Detection in TOEFL Essays

Kevin Kuo
Computer Science
University of Maryland
kvkkuo@gmail.com

Marine Carpuat
Computer Science & UMIACS
University of Maryland
marine@cs.umd.edu

Abstract

This paper describes systems submitted to the Metaphor Shared Task at the Second Workshop on Figurative Language Processing. In this submission, we replicate the evaluation of the Bi-LSTM model introduced by Gao et al. (2018) on the VUA corpus in a new setting: TOEFL essays written by non-native English speakers. Our results show that Bi-LSTM models outperform feature-rich linear models on this challenging task, which is consistent with prior findings on the VUA dataset. However, the Bi-LSTM models lag behind the best performing systems in the shared task.

1 Introduction

In today's globalized world, text in a given language is not always written by native speakers. It is therefore important to evaluate to what degree NLP models and tools developed and evaluated primarily on edited text written and aimed at native speakers port to non-native language. The Metaphor Detection Shared Task at the Second Workshop on Figurative Language Processing offers the opportunity to perform such an evaluation on a challenging genre: argumentative essays written by non-native speakers of English as part of the Test of English as a Foreign Language (TOEFL).

We participate in the TOEFL ALLPOS task, a sequence labeling task where each word in running task is labeled with one of two tags: metaphorical (M) or literal (L). While the best-performing system described in this paper was submitted to other sections of the shared task, we focus on reporting a wider range of results for the TOEFL ALLPOS task.

Context determines whether a word or phrase is being used in a metaphorical sense. Consider an example from the TOEFL dataset: "The world is a huge **stage** and nearly everybody is an **actor**." The words "stage" and "actor" are used metaphorically to analogize the world to a stage and individuals to actors on that stage. A literal usage of these two words would be "The **actor** walked across the **stage**.", because "actor" and "stage" both occur within the context of a theatrical performance, which also matches the context of the sentence.

Beigman Klebanov et al. (2018) establish baselines for metaphor detection on TOEFL essays using feature-rich logistic regression classifiers, and show that use of metaphors is a strong predictor of the quality of the essay. The same year, Gao et al. (2018) establish a new state-of-the-art with a simple Bi-LSTM model on the VUA dataset drawn from multiple genres in the British National Corpus (BNC). Their approach departed from prior models built on linguistically motivated features (Turney et al., 2011; Hovy et al., 2013; Tsvetkov et al., 2014), visual features (Shutova et al., 2016) or learning custom word embeddings (Stemle and Onysko, 2018; Mykowiecka et al., 2018), and showed that contextualized word representations from Bi-LSTM can be more effective.

In this work, we investigate whether Gao et al. (2018)'s findings can be replicated when detecting metaphors in TOEFL essays rather than the BNC. In addition, we attempt to answer the following question: do contextualized word representations from a Bi-LSTM model detect metaphorical word use more accurately than feature-rich linear models? On the one hand, Bi-LSTM sequence labelers have proven quite successful at learning task-specific representations for many NLP problems. On the other hand, text written by non-native speakers of varying proficiency might include more variability that harms the models ability to learn useful contextual representations.

Our results show that Bi-LSTMs with word embedding inputs outperform feature-rich linear classifiers as in prior work, but their performances lag behind that of the top performing submissions in the shared task.

Proceedings of the Second Workshop on Figurative Language Processing, pages 192–196

https://doi.org/10.18653/v1/P17

	Train	Test
Sentences	2741	968
Labeled Tokens	26647	9014
Labeled Tokens (+)	1878	-
Labeled Types	5587	2746

Table 1: TOEFL ALLPOS statistics for the provided training data (train) and the blind evaluation set (test).

2 Task Overview

The goal of the task is to accurately predict whether words are used in a literal or metaphorical sense in a sequence labeling setting. As shown in Table 1, the literal tokens heavily outnumber the metaphorical ones. To account for this imbalance, submissions are evaluated using the F1 score for the positive class (metaphorical). In the table, a "token" refers to a labeled word in the data (not all words are assigned labels/features). We will refer the reader to the shared task description paper for a detailed description of the task.

In addition to metaphor annotations, the corpus comes with pre-extracted features from Klebanov et al. (2015), labeled as *Provided features* in Table 2. These features include unigrams, Stanford POS tags, binned mean concreteness values (Brysbaert et al., 2013), and Topic-Latent Dirichlet Allocation (Blei et al., 2003). Unlabeled tokens are assigned a literal classification and values of zero for all non-word embedding features.

3 System Configurations

3.1 Classifiers

We ran our internal experiments using a simple baseline and two classifier architectures.The implementation, written in Python, will be made publicly available on Github. [1]

Baseline As a baseline (**BL**), we predict the probability $p(w)$ of a word lemma w to be positive (metaphorical) as m_w/c_w, where m_w and c_w are the number of positive occurrences and total occurrences of w respectively in the training data. If $c_w = 0$ (the word was not encountered during training), we automatically assign a negative (literal) prediction.

Linear Classifiers We use a logistic regression (**LR**) classifier implemented using scikit-learn (Pedregosa et al., 2011) with default training settings (LBFGS solver with L2 penalization). We predict a binary classification for each token independently, ignoring other predictions and features in the sequence.

[1] https://github.com/imkevinkuo/metaphor-toefl.

Feature	Dim.	Name
Word embeddings		
GE	1324	GloVe + ELMo vectors
Provided features		
UL	5027	Unigram lemma
P	17	Stanford POS
WN	15	WordNet verb senses
T	100	Topic-LDA
C	34	Concreteness bins
CD	66	Concreteness difference

Table 2: Features available for use.

Bi-LSTM Following Gao et al. (2018), we use a Bidirectional LSTM as a sequence labeler, simply using a feed-forward neural network to make a binary prediction at each time step, using the contextualized representations learned by the Bi-LSTM as input. Predictions are made for each sentence in an essay, independently of the document context. Our experiments are based on the implementation by Gao et al., with modifications to the code in order to apply their model to the TOEFL data and to incorporate different combinations of features.

The LSTMs have a hidden size of 300 units for each direction. Concatenating the forward and backward representations yields a 600-dimensional output. We feed this output through a single-layer (2 units) feedforward neural network and apply a softmax function, which outputs a probability distribution for the two output classes. Dropout is applied to the LSTM input ($p = 0.5$) and from LSTM output to the linear layer ($p = 0.1$). The models are trained using the Adam algorithm, with learning rates of $\eta = 0.005$ and 0.001 for epochs $0 - 10$ and $11 - 20$, respectively.

3.2 Features

We experimented with different input features within each model architecture, which are summarized in Table 2.

We obtain word embedding features for each word type by concatenating GloVe (Pennington et al., 2014) and ELMo (Peters et al., 2018) word embeddings into a 1324-dimensional vector, shown as "GE" in Table 2.

All the other features were provided with the

Model	Features Used	P	R	F1	Test F1
BL	-	64.3	52.6	57.7	54.5
LR	UL	55.2	51.9	53.3	52.4
	UL, P, WN, T, C, CD	58.4	54.1	55.7	50.5
	GE	58.7	63.0	60.5	-
	GE, UL	55.7	60.6	57.9	**56.5**
	GE, UL, WN, CD	61.0	61.7	60.7	-
LSTM	UL, P, WN, T, C, CD	50.6	30.5	38.0	-
	GE	69.3	65.0	66.8	58.2
	GE, UL	73.3	61.9	67.1	**60.9**
	GE, UL, WN, CD	73.8	60.4	66.3	-

Table 3: Summary of results based on 5-fold cross validation on the unmodified training set (P,R,F1) as well as evaluation on the blind test set on CodaLab (Test F1).

TOEFL ALLPOS dataset, which we will refer to as 'provided' features. With the exception of Topic-LDA (T), all of them are represented with one-hot encodings (UL, P) or a vector of binary values (WN, C, CD).

Various combinations of all these features were concatenated together to form the input data on which we trained and evaluated the classifiers described above.

3.3 Data Versions

Default Data We first build classifiers on the data as processed by the organizers, with the provided tokenization and no additional processing.

Since the TOEFL essays are written by non-native English speakers, many sentences contain misspellings or grammatical errors, such as "The problems of the pollution is one of the most ones of this century." We experiment two strategies to address these sources of variability.

Spelling Correction We created a cleaned version of the dataset using the Python *pyspellchecker* library, which finds a given word's minimum Levenshtein distance neighbor in the OpenSubtitles corpus. In total, we replaced 1536 (train) and 492 (test) misspelled tokens in the data.

Error Injection Anastasopoulos et al. (2018) showed that adding synthetic grammatical errors to training data improves neural machine translation of non-native English to Spanish text. To investigate the effect of such methods on metaphor detection, we separately inject the following errors (if applicable) into three copies of each training sentence and append them to the training set:

- RT: Missing determiner (includes articles)
- PREP: Missing preposition
- NN: Flipped noun number

For simplicity, unlike Anastasopoulos et al. we did not randomly replace determiners or prepositions with another member of their confusion set. Instead, we simply removed the word from the sentence.

3.4 Evaluation Settings

When training the logistic regression and Bi-LSTM classifiers, we ran cross-validation ($k = 5$) and used early stopping to select a final test model based on validation loss. We then selected a probability threshold that maximized our F1 score on the validation data before finally making predictions on the test set.

For our baseline model, we used the same model selection technique without early stopping, as there is no 'training' iteration involved in the baseline.

4 Results

4.1 Impact of Classifier and Feature Choice

We first compare classifiers and features when training on the default data. Table 3 includes our internal results averaged across 5-fold cross-validation on the training set, and for a subset of the models, results on the blind evaluation test set taken from official leader board on CodaLab.

The baseline model performs well on both the testing and validation sets, which suggests that the identify of the word is a strong indicator of metaphorical use even before taking context into account, for the TOEFL data as for other genres. Surprisingly, the linear classifiers that did not use word embedding features did not improve over the

baseline, despite the fact that they include the identity of the current lemma (UL). The only models that produced improvements over the baseline on average used GloVe and ELMo embeddings. Additionally, the effect of adding the provided features is inconsistent - in some cases, performance degrades, but in others, it improves.

The difference in F1 score between Bi-LSTM and LR models is primarily due to precision: The Bi-LSTM models that use word embeddings achieve higher precisions than the logistic regression models, while the differences in recall are small. This constrasts with the findings of Gao et al. (2018) on the VUA dataset, where the Bi-LSTM model primarily benefited recall over precision.

The best results overall are obtained with the Bi-LSTM models that use GloVe and ELMo input. Interestingly, adding unigram lemma features (UL) further improves precision and the expense of a small decrease in recall, and overall yields the best F1 both by cross-validation and on the official test set. As expected, Bi-LSTM performance degrades heavily when trained on only the dataset-provided features. Investigating better ways to incorporate these features would be a useful direction for future research. Finally, Table 5 shows our best model's performance, broken down by Penn Treebank POS tags: F1 scores are the highest for verbs and lowest for nouns, mostly due to worse recall for nouns than for verbs.

4.2 Impact of Addressing Spelling and Grammatical Errors

Spell-checking and error injection experiments have an inconsistent impact. As shown in 4, this additional data processing improves the F1 score of the Logistic Regression model most. For the Bi-LSTM, spell-checking the data yields a small F1 improvement when using cross-validation, and no significant difference on the official test set (60.9 vs. 61.0). Injecting artificial errors leads to a small F1 decrease with cross-validation and was therefore not tested on the official test set.

5 Official Submission

Our best submission on the leaderboard is a Bi-LSTM network trained on a spell-checked dataset embedded with GloVe, ELMo, and one-hot unigram lemma vectors. This model yields an F1 score of 0.610, which is slightly below the median score of 0.653.

Model	Data	P	R	F1	Test F1
BL	Base	64.3	52.6	57.7	**54.5**
	Spell	60.6	54.7	57.4	54.3
LR	Base	55.7	60.6	57.9	-
	Spell	60.5	62.4	**61.3**	-
	Errors	58.7	62.5	60.5	-
LSTM	Base	73.3	61.9	67.1	60.9
	Spell	70.5	65.5	67.9	**61.0**
	Errors	70.8	63.5	66.8	-

Table 4: Comparison of averaged 5-fold cross validation results (P,R,F1) on the original text (Base), spell checked data (Spell) and error injected data (Error), as well as evaluation on the blind test set on CodaLab (Test F1). non-BL models use the GE and UL features.

POS	#	% M	P	R	F1
NN	8498	4.8	75.8	54.7	63.5
NNS	4328	2.4	72.2	50.0	59.1
JJ	4024	8.6	83.0	63.8	72.1
VB	2715	16.2	77.8	78.8	78.3
RB	1998	3.2	86.7	68.4	76.5
VBP	1402	6.6	68.2	78.9	73.2
VBG	1188	11.5	73.9	70.8	72.3

Table 5: Evaluation of best Bi-LSTM model per POS tag via cross-validation. We show statistics (count, % metaphoric) for the training set. Only POS tags with more than 1000 occurrences are displayed.

6 Conclusion

In summary, our experiments replicate existing metaphor detection models in the new settings provided by the TOEFL ALLPOS task. Adding GloVe vectors and ELMo contextual embeddings helped push the performance of the logistic regression model over a simple frequency baseline. The use of a Bi-LSTM network in combination with GloVe, ELMo, and one-hot unigram lemma vectors yields the highest performance out of all the models tested. This confirms the benefits of contextual representations learned by the Bi-LSTM for metaphor detection highlighted by Gao et al. (2018) on the VUA dataset. However, the more challenging TOEFL ALLPOS data also shows the limitation of the Bi-LSTM model, which yields smaller improvements over the baseline than on VUA, and lags behind the best systems on the shared task leader board.

References

Antonios Anastasopoulos, Marika Lekakou, Josep Quer, Eleni Zimianiti, Justin DeBenedetto, and David Chiang. 2018. Part-of-speech tagging on an endangered language: a parallel Griko-Italian resource. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2529–2539, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Beata Beigman Klebanov, Chee Wee (Ben) Leong, and Michael Flor. 2018. A corpus of non-native written English annotated for metaphor. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 86–91, New Orleans, Louisiana. Association for Computational Linguistics.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2013. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior Research Methods*, 46(3):904—-911.

Ge Gao, Eunsol Choi, Yejin Choi, and Luke Zettlemoyer. 2018. Neural metaphor detection in context.

Dirk Hovy, Shashank Srivastava, Sujay Kumar Jauhar, Mrinmaya Sachan, Kartik Goyal, Huying Li, Whitney Sanders, and Eduard Hovy. 2013. Identifying metaphorical word use with tree kernels. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 52–57, Atlanta, Georgia. Association for Computational Linguistics.

Beata Klebanov, Chee Wee Leong, and Michael Flor. 2015. Supervised word-level metaphor detection: Experiments with concreteness and reweighting of examples.

Agnieszka Mykowiecka, Aleksander Wawer, and Malgorzata Marciniak. 2018. Detecting figurative word occurrences using recurrent neural networks. In *Proceedings of the Workshop on Figurative Language Processing*, pages 124–127, New Orleans, Louisiana. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Ekaterina Shutova, Douwe Kiela, and Jean Maillard. 2016. Black holes and white rabbits: Metaphor identification with visual features. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 160–170, San Diego, California. Association for Computational Linguistics.

Egon Stemle and Alexander Onysko. 2018. Using language learner data for metaphor detection. In *Proceedings of the Workshop on Figurative Language Processing*, pages 133–138, New Orleans, Louisiana. Association for Computational Linguistics.

Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 248–258, Baltimore, Maryland. Association for Computational Linguistics.

Peter Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 680–690, Edinburgh, Scotland, UK. Association for Computational Linguistics.