Augmenting Neural Metaphor Detection with Concreteness

Ghadi Alnafesah[1,2] and **Harish Tayyar Madabushi**[1] and **Mark Lee**[1]

[1] University of Birmingham, UK
(gxa713, H.TayyarMadabushi.1, m.g.lee)@bham.ac.uk

[2] Qassim University, KSA
gm.alnafesah@qu.edu.sa

Abstract

The idea that a shift in concreteness within a sentence indicates the presence of a metaphor has been around for a while. However, recent methods of detecting metaphor that have relied on deep neural models have ignored concreteness and related psycholinguistic information. We hypothesis that this information is not available to these models and that their addition will boost the performance of these models in detecting metaphor. We test this hypothesis on the Metaphor Detection Shared Task 2020 and find that the addition of concreteness information does in fact boost deep neural models. We also run tests on data from a previous shared task and show similar results.

1 Introduction

The automatic detection and processing of metaphor is an ongoing challenge for true deep semantic understanding of natural language text. Metaphors often convey unrelated concepts to their literal meaning and the meaning of metaphor involves more than just its words meaning, but it incorporates the whole context with a wider knowledge of their conceptual domain.

Traditional methods of metaphor detection that do not make use of neural networks have used concreteness scores to improve metaphor detection (Turney et al., 2011; Tsvetkov et al., 2013) . However, neural models that use distributional semantics (i.e. word embeddings: Mikolov et al. (2013)) have shown promising and often state-of-the-art results in a range of NLP tasks and have recently produced promising results in metaphor detection (Mao et al., 2018; Mishra et al., 2019; Rei et al., 2017). These models, however, focus on the textual information provided by the word embeddings and do not further explore the use and effect of combining other lexical information. This paper reports the result of combining neural networks with a lexical resource for measuring concreteness for word-level metaphor detection.

Despite the success of deep neural models, we hypothesise that they do not have access to concreteness information with their structure. To test this, we explicitly add concreteness information to deep neural models and compare their performance with and without this information. Our experiments show that deep neural models, like more traditional models, do benefit from concreteness information.

2 Related Work

Early work, by Turney et al. (2011) on the use of concreteness to detect metaphor made use of the relatively small MRC psycholinguistic dataset (Coltheart, 1981) for concreteness scores. Their work uses a logistic regression model to detect the metaphoricity of adj-noun pairs in the TroFi dataset (Birke and Sarkar, 2006). Subsequently, Tsvetkov et al. (2013) made use of the same MRC dataset to detect subject-verb-obj metaphors from TroiFi dataset. They also train a supervised logistic regression classifier on English triples and test on a Russian dataset. Köper and Schulte im Walde (2017b) extend this work by using a significantly larger dataset (Brysbaert et al., 2014) of concreteness ratings and propagating the concreteness rating to phrases using word2vec (Mikolov et al., 2013). Their experiments use Leong et al. (2018)'s Logistic Regression classifier on VUAMC for verbs using ten-fold cross-validation process.

The context that a word occurs in plays an important role in metaphor detection (Klebanov et al., 2014). Words and phrases often convey very different meanings in different contexts. Consider the phrase "*cut down*" in the sentence "*She cut down his advances with her words.*" In ab-

Proceedings of the Second Workshop on Figurative Language Processing, pages 204–210

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

sence of the context, it is not clear that "*cut down*" is metaphorical. Many supervised learning approaches, including those described above, utilise bag of words methods, thus focusing on sets of features which do not capture context. Those that do consider context, do so only to a small extent, for example by focusing only on specific sentence constructs like *adj-noun* pairs (Bizzoni et al., 2017) or *subj-verb-obj* (Tsvetkov et al., 2013).

Given the importance of context and the power of neural models in capturing context, it was only natural to use deep neural models for metaphor detection. Gao et al. (2018) make use of deep neural networks to detect metaphor with significant success across multiple datasets including VUAMC. In particular they use Bidirectional Long Short Term Memory networks (Bi-LSTM) that capture relations in both directions for word-level metaphor classification with word-embeddings as input.

Other work on using concreteness and similar psycholinguistic features for metaphor detection include that by Bulat et al. (2017) who combined concreteness with property norms to formulate representations. Ljubešić et al. (2018) combine imageability scores with concreteness for cross-lingual metaphor detection and Dunn (2015) make use of abstractness.

This paper reports the result of applying concreteness score to individual words in the token-level metaphor classification for the Metaphor Detection Shared Task competition 2020. We build on Gao et al. (2018)'s sequence labelling network by adding concreteness scores to individual words.

The arrival of deep neural networks has meant that psycholinguistic features are no longer explicitly considered and, as mentioned in Section 1, we hypothesise that deep neural networks do not have access to this information. In this work, we show that this is the case and that access to this information improves the accuracy of deep neural networks by testing on multiple datasets.

3 Generalising Concreteness Scores

We used the resource created by Brysbaert et al. (2014) for concreteness scores. This is a list of about 40,000 English words rated for concreteness between 1 to 5 where 1 is most abstract and 5 is most concrete. As an illustration, "*wood*" has a rating of 4.85, "*counterargument*" a rating of 2.17 and "*conventionalism*" 1.18.

Before we can use concreteness scores for metaphor detection, we need a way of handling those words in our dataset that do not have corresponding concreteness scores in the concreteness lexical resource created by Brysbaert et al. (2014). The most obvious solution is to set the concreteness scores of these words to 0. However, the fact that a large number of words in our dataset do not have corresponding concreteness scores makes this impractical.

To get around this, we use the concreteness values available to train a Support Vector Machine. We use BERT (Devlin et al., 2018) embeddings as features to the SVM and the rounded up concreteness values as output classes. So as to use BERT embeddings as input to an SVM, we extract static, non-contextual BERT embeddings. We choose to use BERT, as opposed to static embedding like word2vec, due to BERT's unique tokenizer that allows for the generation of embeddings for all words in our dictionary. We use the following hyperparameters for the SVM: hidden layer sizes 100, activation identity, solver adam, alpha 0.0001, batch size auto, learning rate adaptive, learning rate init 0.001, powert 0.5, max iteration 200, shuffle True, random state None, tol 0.0001, verbose False, warm start False, momentum 0.9, nesterovs momentum True, early stopping False, validation fraction 0.1, beta1 0.9, beta2 0.999, epsilon 1e-08, and niter_no_change 10.

4 Neural Metaphor Detection with Concreteness

We use Gao et al. (2018)'s sequence labeling model as the baseline and modify it to include a concreteness rating as follows. For every input word x_i we modify w_i, the 300-D GloVe pretrained embedding for x_i, with the concreteness class assignment c_i of x_i. This results in a 301-D representation $[w_i : c_i]$ for each of the input words. These representations of words are fed to the sequence labeling model, which consists of a Bi-LSTM which generates a contextual representation of each word. These are then fed to feedforward neural networks which predict a label for each word. Figure 1 provides an illustration of the sequence labeling model, wherein the Bi-LSTM is represented by pink blocks and the blue blocks represent the feedforward neural networks.

We also test appending the probabilities of each of the four concreteness classes output by the

SVM. In this case the 300-D pre-trained representation w_i is concatenated with a vector p_i of length four, where each digit represents the probability of this word belonging to the output class 1, 2, 3 or 4 respectively. This results in a 304-D representation $[w_i : p_i]$ for each word. This method of using the probability distribution is unlike previous methods that have used a single concreteness score. We use the concreteness scorers generated by our SVM model even when a word and the corresponding concreteness score is included in the dataset provided (Brysbaert et al., 2014) and used as the training data for our SVM. We find that the addition of probabilities is far more effective than the addition of a single score possibly because this provides more of a signal for the model to pick up on (4 features not 1).

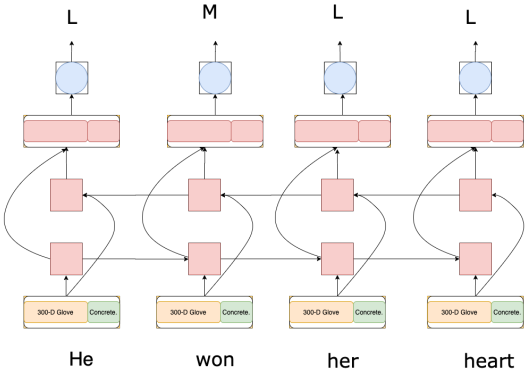

Figure 1: The sequence classification model architecture used in the experiment.

Importantly, if pre-trained embeddings (in our case GloVe) contained concreteness information, the explicit addition of this information by means of appending it to the embeddings should not improve the performance of a well trained Bi-LSTM model as such models are capable of extracting relevant information from their input. An improvement in performance with the addition of concreteness information would imply that such information is not contained in the pre-trained embeddings we use.

5 Results

The metaphor detection shared task allowed multiple submissions and we use this to evaluate different models, both with and without concreteness scores. We present this comparative analysis of our models first before describing our performance in Section 5.2. We also test our models on the previous shared and present these results in Section 5.3.

5.1 Comparitive Analysis

Table 1 summarises the results of our experiments on the VUA ALLPOS dataset. The results on the Shared Task data without concreteness rating is considered the baseline for measuring the model's performance. "Single Class Rating" refers to the model where a single number representing the class of the word was appended to the word's embedding, "Probability Rating" refers to the model where the probability for each class output by the SVM was was concatenated to the word embeddings.

Experiment	Precision	Recall	F1
Gao et al. (2018) with Shared Task Dataset	64.9%	48.9%	55.8%
Single class rating	60.3%	53.7%	56.8%
Probability rating	63.6%	52.9%	57.8%
Probability with rating 2 layers	65.5%	53.2%	58.7%
Probability rating with 3 layers	65.3%	54.8%	**59.6%**

Table 1: A comparison of models with and without concreteness.

Interestingly, the model that used the probabilities of each of the output classes performs the best. Further hyperparameter optimisation (by increasing the number of layers by one) increased F1 score to reach 59.6%. Modifying other hyperparameters did not improve performance. The values of the hyperparameters we use are: 10 epochs, hidden size of 300, batch size of 64, learning rate of 0.005, 1 hidden layer, and LSTM dropouts of 0.5 0 and .1 for input hidden and output layers respectively. So as to ensure that the addition of concreteness rankings is not simply introducing noise, we plot the loss for training and validation which is presented in Figure 2. A subjective analysis of these results is presented in Section 6.

5.2 Shared Task Results

We test our model on the VU Amsterdam Metaphoricial Corpus (VUAMC) by participating in the The Second Shared Task on Metaphor Detection for VUA AllPOS dataset. Our performance

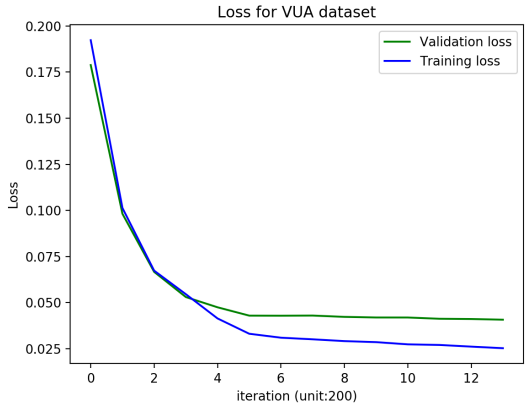

Figure 2: The training and validation loss for the sequence classification model.

on the task is show in Table 2

Rank	Team	F1
1	DeepMet	76.9%
2	xchenets	73.4%
3	meta-phor	73.0%
	...	
13	**UoB Team**	59.6%
14	eduardgzaharia	55.2%

Table 2: Our performance on the shared task.

The lackluster performance on the task can possibly be attributed to our use of static embeddings as opposed to the more powerful contextual pre-trained embeddings such as BERT. We intend to integrate concreteness into BERT models for metaphor detection in our future experiments (Section 7).

5.3 Further Experiments with Verbal Metaphor Detection

In addition to participating in the shared task we also experiment with the Gao et al. (2018)'s version of VUAMC dataset published by Leong et al. (2018) for 2018 Metaphor Shared Task. It should be noted that Gao et al. (2018) modify the task of metaphor detection to one of classification. While the shared task required the classification of metaphor at the word-level, Gao et al. (2018) provide a verb and a sentence containing that verb as input and required classifying that verb into either "Metaphor" or "Not Metaphor".

Once again, we use our reproduced results[1] of the target classification model by Gao et al. (2018) as our baseline and augment it with concreteness scores as we did for this year's tasks. The classification model, like the sequence labeling model feeds word representations to a Bi-LSTM which generates a contextual representation of each word. Unlike in the sequence labeling model, the BiLSTM includes attention and these representations are concatenated and fed to a single feedforward neural network which predicts the label of the verb. Figure 3 provides an illustration of the classification model, wherein the Bi-LSTM is represented by pink blocks, the concatenated representation as the red square and the blue block represents the feedforward neural network. The coloured in circle represents the (highlighted) verb of interest in the sentence.

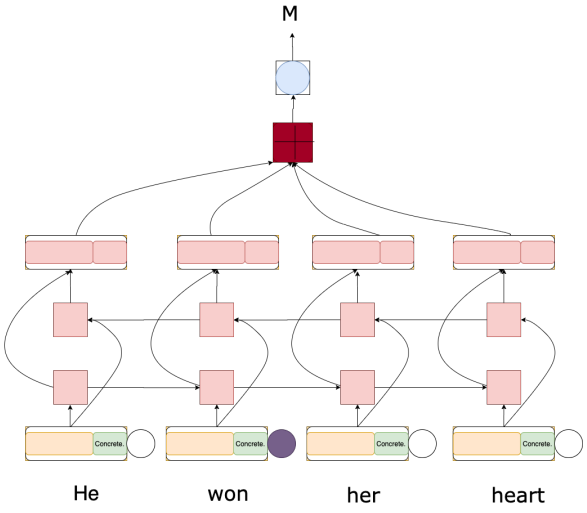

Figure 3: The classification model used for verb metaphor detection.

The values of the hyperparameters we use: 20 epochs, hidden size of 128, batch size of 16, learning rate of 0.01, hidden layers 1, and LSTM dropouts of 0.5 0.0 and 0.2 for input hidden and output layers respectively. The results of our experiments are presented in Table 3.

The classification model also has ELMo Peters et al. (2018) embeddings concatenated to the GloVe embeddings and concreteness score. The incorporation of ELMo embeddings ensures that we capture contextual information. The fact that the addition of concreteness to contextual embeddings shows improvement implies that contextual embeddings do not have access to concreteness information either.

[1] Gao et al. (2018) note that the model that they make available does not include the final hyperparameters used to generate their reported results.

Experiment	Precision	Recall	F1
Gao et al. (2018) classification *reproduced*	55.85%	49.80%	52.65%
Single class rating	57.93%	44.57%	50.38%
One-hot encoding	54.66%	52.18%	55.41%
Probability rating	52.02%	62.86%	56.92%
Probability rating + hyperparameter tuning	54.21%	62.46%	**58.04%**

Table 3: Summary of the experiments results on the classification task.

6 Analysis and Discussion

The training data from the VUAMC dataset has 181,501 tokens, 19,177 of which are labeled metaphor with 162,324 labeled literal. An exploration of the results shows that the most frequently occurring words in the dataset are prepositions. The word *of*, for example, occurs 4,638 but is labeled as a metaphor only 151 times. *With*, on the other hand, appears 995 times and is labelled as a metaphor 620 times and *up* is labelled 137 times as metaphor out of 335 occurrences. Table 5 shows a couple of most frequent words in the dataset along with the number of true positives and false negatives. It appears that prepositions appear so frequently that the distinction between their literal and metaphorical sense is hard to distinguish. For example, the model incorrectly classified the word *of* as literal in the the sentence "*Francesca Simon describes some of the pitfalls and how to avoid them.*"

In addition, prepositions also appear as part of phrases, such as in "*some of the*" making it harder still to classify them correctly. Often, the meaning of a phrasal verb differs significantly from that meaning of its parts. Additionally, concreteness of each of the individual parts is also different from that of the phrase. For example, in the sentence "*The use of role play, incorporating previously discussed difficulties (i.e. homework assignment session 4) in real or set up situations provide an opportunity for testing these skills.*", the overall meaning of the phrase *set up* is different from the meaning of *set* and *up*. Additionally, we were able to successfully classified *set* as a metaphor, but failed to classify *up* as a metaphor in this context.

The partial sentence "*real or set up situations*" has the following information: The word *real*, a literal, has concreteness rating equals to 2 is correctly classified as literal. The word *or* is correctly labelled as literal has concreteness rating of 1. The word *set* is correctly classified as metaphor has concreteness rating of 3. Followed by the word *up* which is incorrectly labelled as literal has concreteness of 3. Lastly, the word *situations* is incorrectly classified as literal has concreteness rating of 2. The noticeable shift in concreteness from rating 1 to 3 for *or* and *set* could lead to successfully classifying set as a metaphor but failed to classify *up* as also a metaphor, although the two forms the meaning of the phrasal vary, because *up*'s rating is not very far from *set*'s rating. A similar error occurs when classifying the phrasal verb "*put up with*" in the sentence "*they also have to put up with the heaviest police presence .*"

Each word meaning by itself differs from the meaning of the whole phrase. *Put* means to place something physically, *up* means the position *up* and *with* mean accompanied by someone or something; however, these three together refer "*to accept an unpleasant situation, something or someone* (willingly or not)." As for their and their near context degrees of abstractness are as follows: The word *to* is correctly labelled as literal has rating equal to metaphor, *put* is correctly labelled as literal has rating 2 as its concreteness rating, followed by the word *up* that is correctly labelled as literal and has a rating of 3. Next is the word *with* that is incorrectly classified as literal has concreteness of 2 and lastly, the word *the* is correctly classified as literal has concreteness of 1. Since there is no drastic shift in concreteness or their senses, the model fails to spot the hits and labels them all as literal.

Of the 15,439 unique tokens in the dataset, 7527 tokens appear exactly once. For example, "*There were others , but Lucy never disclosed any of them to us*" the word *disclosed* is labelled as metaphor but incorrectly classified as literal. There are two interpretations for this sentence. The metaphorical sentence talks about uncovering "*people's identities*" Lucy knew when referencing *others* and *them*, or could literally talks about uncovering of "*secrets*" Lucy hides, which are referenced by *them* and *others*. As for the concreteness ratings for the sentence's words, the rating range between 1 and 2, other than *Lucy* that has rating of 4; therefore, we could say that the rating did not help to

The sentence	The label	The Predicted Label	Concreteness Rating
And they told it without onscreen questioning , though the programme is *skilfully* structured to give it a coherence it might have lacked .	0	0	1
The burn threads a wild and inhospitable crevice of the hills , where the wind blows cold and the sense of *isolation* grows with each lonely mile .	1	0	2
Although that is the position in law , the court emphasised that as a matter of sense a tenant should first complain to the landlord before *exercising* the right to prosecute .	1	1	2

Table 4: Sample of sentences that contain words used only once throughout the dataset, their label, predicted label and their concreteness rating.

Word	True Positive	False Negative	Count
of	3	148	4638
to	538	239	3731
in	1198	285	2811
with	510	110	995
go	24	40	258

Table 5: Sample of words with the highest word counts in the dataset, and their counts for how many times the model correctly classified them as metaphors or failed by classifying them as literal.

clarify the meaning. To better understand the intended meaning (literal or metaphorical), this ambiguous sentence needs more context. The same can be said about the word *corruption* in "*Bribery and corruption!*" This sentence word's concreteness ratings are (2, 1, and 2) respectively; thus, to correctly classify *corruption* as metaphor, more context is required. Table 4 show more sentences containing words that appeared only once along with their labels, predicted labels and concreteness classes.

7 Conclusion and Future Work

This paper reports the results of providing deep neural models with concreteness information by appending a measure of concreteness to word embedding for all content words. Our hypothesis is that explicitly adding a concreteness rating to the word representation will boost the neural network performance in detecting metaphors as neural models do not have access to this information. We tested two representations of concreteness, one as a scale and the other is class probabilities using the VUA ALLPOS data from the Second Metaphor Detection Shared Task 2020 and data from the First Metaphor Detection Shared Task 2018 and find that this information does boost performance in all cases.

We plan on testing the effectiveness of incorporating other psycholinguistic information, such as imageability, into deep neural models so as to establish their impact on metaphor detection. We also intend to incorporate these features into contextual pre-trained models, such as BERT (Devlin et al., 2018) as context is critical to identifying metaphor. In this current work, BERT pre-trained representations were used only in training an SVM and not in the Bi-LSTM that detects metaphor.

We also intend to use more complex models to expand concreteness, imageability and other such features to a larger vocabulary. These models will be designed to perform classification better and also capture context so as to better identify the concreteness of words in context. Finally, we intend to extend our work to include phrases a significant source of errors in this task.

References

Julia Birke and Anoop Sarkar. 2006. A clustering approach for nearly unsupervised recognition of nonliteral language. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.

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.

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.

Luana Bulat, Stephen Clark, and Ekaterina Shutova. 2017. Modelling metaphor with attribute-based semantics. *Proceedings of the 15th Conference of the*

European Chapter of the Association for Computational Linguistics.

Max Coltheart. 1981. The mrc psycholinguistic database. *The Quarterly Journal of Experimental Psychology Section A*, 33(4):497–505.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jonathan Dunn. 2015. Modeling abstractness and metaphoricity. *Metaphor and Symbol*, 30(4):259–289.

Ge Gao, Eunsol Choi, Yejin Choi, and Luke Zettlemoyer. 2018. Neural metaphor detection in context. *arXiv preprint arXiv:1808.09653*.

Beata Beigman Klebanov, Ben Leong, Michael Heilman, and Michael Flor. 2014. Different texts, same metaphors: Unigrams and beyond. In *Proceedings of the Second Workshop on Metaphor in NLP*, pages 11–17.

Maximilian Köper and Sabine Schulte im Walde. 2017b. Improving verb metaphor detection by propagating abstractness to words, phrases and individual senses. In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 24–30, Valencia, Spain. Association for Computational Linguistics.

Chee Wee Leong, Beata Beigman Klebanov, and Ekaterina Shutova. 2018. A report on the 2018 vua metaphor detection shared task. In *Proceedings of the Workshop on Figurative Language Processing*, New Orleans, LA.

Nikola Ljubešić, Darja Fišer, and Anita Peti-Stantić. 2018. Predicting concreteness and imageability of words within and across languages via word embeddings. *arXiv preprint arXiv:1807.02903*.

Rui Mao, Chenghua Lin, and Frank Guerin. 2018. Word embedding and wordnet based metaphor identification and interpretation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics (ACL).

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2019. Author profiling for hate speech detection. *arXiv preprint arXiv:1902.06734*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Marek Rei, Luana Bulat, Douwe Kiela, and Ekaterina Shutova. 2017. Grasping the finer point: A supervised similarity network for metaphor detection. *arXiv preprint arXiv:1709.00575*.

Yulia Tsvetkov, Elena Mukomel, and Anatole Gershman. 2013. Cross-lingual metaphor detection using common semantic features. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 45–51.

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