Procesamiento del Lenguaje Natural, Revista nº 64, marzo de 2020, pp. 53-60 recibido 21-11-2019 revisado 07-01-2020 aceptado 09-01-2020

Metaphoricity Detection in Adjective-Noun Pairs

Detección de Metaforicidad en Pares Adjetivo-Sustantivo

Andrés Torres Rivera, Antoni Oliver, Marta Coll-Florit
Universitat Oberta de Catalunya
{atorresrive, aoliverg, mcollfl}@uoc.edu

Abstract: In this paper we propose a neural network approach to detect the metaphoricity of Adjective-Noun pairs using pre-trained word embeddings and word similarity using dot product. We found that metaphorical word pairs tend to have a lower dot product score while literal pairs a higher score. On this basis, we compared seven optimizers and two activation functions, from which the best performing pairs obtained an accuracy score of 97.69% and 97.74%, which represents an improvement of 6% over other current approaches.
Keywords: NLP, Metaphor, Word Embeddings, Deep Learning

Resumen: En este artículo proponemos un acercamiento mediante redes neuronales para la detección de la metaforicidad de pares Adjetivo-Sustantivo empleando *word embeddings* pre-entrenados y similitud de palabras mediante el producto escalar. Encontramos que los pares de palabras metafóricos tienden a tener un producto escalar bajo mientras que los pares no metafóricos un resultado más alto. Bajo este supuesto, comparamos siete optimizadores y dos funciones de activación, de las cuales los pares con mejor desempeño obtuvieron una exactitud de 97.69% y 97.74%, que representa una mejora de 6% sobre otros enfoques actuales.
Palabras clave: PLN, Metáfora, Word Embeddings, Aprendizaje Profundo

1 Introduction

The automatic detection of figurative language is one of the most challenging tasks in Natural Language Processing (NLP). Specifically, metaphor is the most studied process, as it is omnipresent in natural language text and therefore it is crucial in automatic text understanding (Shutova, 2010).

According to the Conceptual Metaphor Theory (Lakoff and Johnson, 1980), a metaphor represents a mapping of abstract concepts (target domain) to more concrete or tangible phenomena (source domain), as in the following examples, which are instances of the conceptual metaphor TIME IS MONEY:

You're wasting my time.

This gadget will save you hours.

Two main kinds of metaphor can be distinguished: *conventional* metaphors, which are commonly used in everyday language (as the examples above), and *novel*, *literary*, *creative* or *unconventional* metaphors, which surprise our imagination.

The study of metaphor is a prolific area of research in Cognitive Linguistics, being the Metaphor Identification Procedure (MIP) (Pragglejaz Group, 2007) and its derivative MIPVU (Steen et al., 2010) the most standard methods for manual metaphor detection. Moreover, in the area of Corpus Linguistics, some methods have been developed for annotation of metaphor in corpora (Shutova, 2017; Coll-Florit and Climent, 2019).

In reference to NLP, methodologies for automatic processing of metaphors can be classified into three main categories (Veale, Shutova, and Klebanov, 2016):

- *Corrective approaches*, the earliest ones, where metaphors are considered as a deviation of literal language that must be corrected.
- *Analogical approaches* where metaphors are viewed as some cross-domain transfer of semantic structure.
- *Schematic approaches* where each metaphorical expression is understood as an instance of a more general metaphorical schema.

ISSN 1135-5948. DOI 10.26342/2020-64-6

All these approaches have the following points in common: (1) assume the existence of a literal (or at least normative) meaning of words; (2) assume that some form of structural mapping is required to obtain an interpretation of the metaphor; and (3) assume that metaphor itself is a unit of conceptual representation.

According to Shutova (2010), there are two main tasks in the automatic processing of metaphors:

- *Metaphor recognition*: distinguishing between literal and metaphorical language in a text.
- *Metaphor interpretation*: identifying the intended literal meaning of a metaphorical expression.

Recently, techniques for metaphor recognition are shifting from classical machine learning techniques, as classifiers and decision trees, to the use of more advanced Artificial Intelligence techniques, as neural networks.

The main goal of this paper is to present a new model for metaphor recognition, and specifically for metaphoricity detection of adjective-noun pairs, from a neural network approach. Below we describe the main related works (section 2). Next we present our methodology and model (section 3) and the main results (section 4). We finish with the discussion and our overall conclusions (sections 5 and 6).

2 Related work

Current approaches regarding metaphor recognition include the works of Rosen (2018), Wu et al. (2018) and Mu, Yannakoudakis, and Shutova (2019), which focus on the detection of metaphorical instances in general corpora. Our work focuses on a different task within the scope of metaphor recognition that consists on detecting the metaphoricity of adjective-noun (AN) pairs in English as isolated units. Current approaches on this task include the works by Turney et al. (2011), Tsvetkov et al. (2014), Gutierrez et al. (2016) and Bizzoni, Chatzikyriakidis, and Ghanimifard (2017).

In relation to metaphor recognition in general corpora, Rosen (2018) developed an algorithm using deep learning techniques that uses a representation of metaphorical constructions in an argument - structure level. The algorithm allows for the identification of source-level mappings of metaphors. The author concludes that the use of deep learning algorithms including construction grammatical relations in the feature set improves the accuracy of the prediction of metaphorical source domains.

Wu et al. (2018) propose to use a Convolutional Neural Network - Long-Short Term Memory (CNN-LSTM) with a Conditional Random Field (CRF) or Softmax layer for metaphor detection in texts. They combine CNN and LSTM to capture both local and long-distance contextual information to represent the input sentences.

Some authors (Mu, Yannakoudakis, and Shutova, 2019) argue that using broader discourse features can have a substantial positive impact for the task of metaphorical identification. They obtain significant results using document embeddings methods to represent an utterance and its surrounding discourse. With this material a simple gradient boosting classifier is trained.

With regard to metaphoricity detection in AN pairs, the work of Turney et al. (2011) is based on the hypothesis that metaphorical word usage is correlated with the degree of abstractness of the context of a word. The idea comes from research in Cognitive Linguistics that views metaphor as a cognitive strategy to map knowledge between two domains: one of the domains is familiar, well-understood or concrete; and the other domain is unfamiliar, less understood or more abstract. They present an algorithm to classify a word sense in a given context as literal (denotative) or metaphorical (connotative) and evaluate the algorithm in a set of annotated AN phrases. One of the strengths of the approach is that it can generalize to new words outside the training data.

In Tsvetkov et al. (2014) a model to discriminate whether a syntactic construction has a literal or metaphoric sense is presented. The model uses lexical semantic features of the words in the construction. One of the advantages of the model is that it can be transferred to other languages by pivoting through a bilingual dictionary. They work with two syntactic constructions: subject-verb-object (SVO) and, like in our study, adjective-noun (AN) tuples.

In Gutierrez et al. (2016) a test case for compositional distributional semantic models (CDSMs) is presented. The authors propose a method to learn metaphors as lineal transformations in a vector space. They show that modeling metaphor explicitly within a CDSM can improve the resulting vector representations. As metaphors show a high degree of systematicity, it is possible to learn linear transformations for the representation of metaphorical mappings for adjetives in the same semantic domain.

Finally, in Bizzoni, Chatzikyriakidis, and Ghanimifard (2017) a single neural network with pre-trained vector embeddings is used to identify metaphors in AN pairs. The system is able to provide a metaphoricity score as an output. Table 1 presents the accuracy score of the current approaches in AN metaphoricity detection which establishes a current performance of 91% in accuracy.

The approaches proposed by Turney et al. (2011) and Tsvetkov et al. (2014) implement feature engineering (FE) using small annotated (Ann.) datasets. Currently, Gutierrez et al. (2016) and Bizzoni, Chatzikyriakidis, and Ghanimifard (2017) opt for approaches that do not implement FE, instead both present models trained using embeddings: a distributional semantic model (DSM) in the first case, and word2vec in the second case. In both instances the training and testing data was generated using the AN corpus compiled by Gutierrez et al. (2016).

	A	**FE**	**Ann.**
Turney et al. (2011)	0.79	Y	100
Tsvetkov et al. (2014)	0.85	Y	200
Gutierrez et al. (2016)	0.81	N	8592
Bizzoni et al. (2017)	0.91	N	8592

Table 1: Accuracy score comparison of metaphorical AN pairs detection

3 *Methodology*

Our study is based on the annotated AN pairs corpus presented by Gutierrez et al. (2016), which is composed by 8,592 word pairs that are a combination of 23 unique adjectives and 3,418 unique nouns. This corpus can be divided in two subsets: one composed by 4,601 metaphoric pairs and another composed by 3,991 literal or non metaphorical pairs with an interannotator reliability of $\kappa = 0.80$ and a standard error (SE) of 0.2. Both subsets include cases of metaphoric pairs with each of the 23 adjectives, but in the case of nouns the metaphoric subset contains 2,027 unique nouns whereas the non metaphoric subset 1,547 unique nouns.

Gutierrez et al. (2016) focused on adjectives that function as source domain words in productive conceptual metaphors (CM). Some examples of this kind of CM found in the AN corpus include: *bright day*, *rough character*, *heavy expansion*, and *bitter competition*. As shown in Table 2, the 23 adjectives were divided in eight source-domain categories.

Source	**Adjectives**
Temperature	Cold, heated, icy, warm
Light	Bright, brilliant, dim
Texture	Rough, smooth, soft
Substance	Dense, heavy, solid
Clarity	Clean, clear, murky
Taste	Bitter, sour, sweet
Strength	Strong, weak
Depth	Deep, shallow

Table 2: Categories of the 23 adjectives that compose the AN corpus

We used pre-trained word vectors that were trained using part of the Google News dataset. This model contains 300-dimensional vectors with a context window size of 5 (Mikolov et al., 2013a; Mikolov et al., 2013b; Mikolov, Yih, and Zweig, 2013; Le and Mikolov, 2014). We opted to use these vectors in order to reproduce the process followed by Bizzoni, Chatzikyriakidis, and Ghanimifard (2017).

3.1 Dot product as a similarity measure

Within an Euclidean space, the dot product (Equation 1) is the result of multiplying the magnitudes of two equal-length vectors and the cosine of the angle between them. The result of this operation is a scalar value that can be interpreted as the similarity between to vectors: vectors that have a low score tend to be less similar while vectors that have a higher score tend to be more similar. Word embeddings are n-dimensional vectors that contain semantic and lexical information from all the words that compose the training vocabulary. Computing the dot product between two given word vectors might indicate the similarity relation that exists between them, as shown by Mikolov, Yih, and Zweig (2013), inasmuch as similar words tend to appear near each other within a vector space.

$$A \cdot B = ||A||\,||B|| \cos\theta \tag{1}$$

After computing the dot product for each AN pair, we observed that metaphorical pairs presented a mean result of 0.8548 with an standard deviation (SD) of 0.6865, while the mean result for literal pairs was 1.2545 with a SD of 0.8418. As shown in Figure 1, metaphorical AN pairs (blue) tend to have a lower dot product score while literal AN pairs (orange) have a higher score, which might indicate that literal AN pairs tend to be more similar, and metaphorical AN pairs are combinations of words that are less similar.

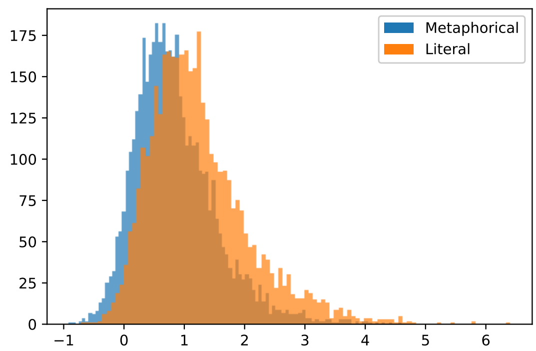

Figure 1: Dot product comparison of metaphorical and literal AN pairs

This values are observed across all sources, Table 3 shows the mean dot product score by source and tag (Literal or Metaphoric). Only the source *Strength* has a higher metaphoric mean dot product in comparison with its literal counterpart. In all the other sources the literal AN pairs have a higher mean dot product result. In some cases such as the sources *Depth* and *Texture* this score almost doubles the value obtained by the metaphoric AN pairs.

The five highest dot product scores were obtained by literal AN pairs belong to the *Temperature* source, such as: *icy snow, icy arctic, icy blizzard, cold child* and *icy precipitation*. On the other hand, the lowest dot product score where mostly obtained by metaphorical AN pairs that belong to different sources, such as: *clean datum, bitter identification, brilliant parent* and *rough customer*, with one instance of the literal pair *shallow outfit*.

Source	Tag	Mean	SD
Clarity	Lit.	0.957033	0.705107
	Met.	0.733197	0.544971
Depth	Lit.	1.564873	0.865550
	Met.	0.778630	0.553173
Light	Lit.	1.276814	0.859143
	Met.	0.824224	0.647932
Strength	Lit.	0.628803	0.433746
	Met.	0.799933	0.583103
Substance	Lit.	1.019069	0.592838
	Met.	0.650521	0.548152
Taste	Lit.	1.996791	0.884432
	Met.	1.270854	0.887818
Temperature	Lit.	1.352197	0.938974
	Met.	0.993028	0.770670
Texture	Lit.	1.209835	0.585611
	Met.	0.699859	0.580966

Table 3: Mean dot product score and standard deviation (SD) by source and tag

3.2 Model description

Our model consists of a variation of the first architecture proposed by Bizzoni, Chatzikyriakidis, and Ghanimifard (2017). Under this architecture, a network is a generalization of the additive composition model (Equations 2 and 3) proposed by Mitchell and Lapata (2010), but using a weight matrix W that modifies all feature dimensions at the same time.

$$\mathbf{p} = (\mathbf{u}, \mathbf{v}; \theta) \tag{2}$$

$$\mathbf{p} = W_{adj}^T \mathbf{u} + W_{noun}^T \mathbf{v} + b \tag{3}$$

This approach can be implemented by concatenating word vectors before feeding them to a neural network. In this case, the parameter function is defined according to equations (4) and (5):

$$W = \begin{bmatrix} W_{adj} \\ W_{noun} \end{bmatrix} \tag{4}$$

$$\mathbf{p} = f_\theta(\mathbf{u}, \mathbf{v}) = W^T \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} + b \tag{5}$$

Using the observed scores of the dot products of the AN pairs, we propose a variation of the multiplicative model presented by Mitchell and Lapata (2010), where instead of computing tensor multiplication we compute the dot product of each AN pair using their embeddings. With this modification we obtain the projection of vector $\mathbf{u}$ over $\mathbf{v}$ (Equation 6), and thus the network is fed a scalar value that can be interpreted as the similarity relation that exists between a given word vector pair.

$$\mathbf{p} = f_\theta(\mathbf{u}, \mathbf{v}) = W_{adj}^T \mathbf{u} \cdot W_{noun}^T \mathbf{v} + b \quad (6)$$

To evaluate the performance of our model we compared the accuracy score of 7 optimizers (Adam, Nadam, Adamax, Adagrad, Adadelta, Stochastic Gradient Descent [SGD] and RMS Prop) with ReLu and linear function as activation functions. In all cases we set binary cross-entropy as the loss function, and used a 10 K-fold cross validation to obtain the mean accuracy score of each optimizer-activation pair. After performing this evaluation we proceeded to evaluate the best performing models to compare their mean accuracy error, precision, recall and f1-score. The model was trained using the same parameters proposed by Bizzoni, Chatzikyriakidis, and Ghanimifard (2017), i.e. it was trained for 20 epochs using 500 examples for training and the rest for testing.

4 Results

After training each model we calculated the mean accuracy using 10 K-Fold cross validation. As shown in Table 4, the set of optimizers using the linear activation function obtained a mean of 97% accuracy. The highest score was obtained by the model trained using the Adagrad optimizer, which obtained an accuracy equal to 97.69%, while the lowest scoring model was the one trained using SGD with an accuracy equal to 69.97%.

Optimizer	**A**	**SD**
Adam	97.58	0.4622
Nadam	97.62	0.5270
Adamax	97.56	0.5799
Adagrad	**97.69**	0.5182
Adadelta	97.65	0.5798
SGD	69.49*	15.0106
RMS Prop	97.48	0.3358

Table 4: Linear Function Accuracy Score (A) and Standard Deviation (SD)

The second set of optimizers was trained using ReLu as activation function. In this case the overall scores where around 97%, the highest accuracy score was 97.74%, obtained by the model trained using Nadam+ReLu. We can also observe a considerable improvement in the case of SGD+ReLu, which obtained an accuracy score of 92.16%. This represents an improvement of 22.67% in comparison with its SGD+linear function equivalent.

Optimizer	**A**	**SD**
Adam	97.63	0.4012
Nadam	**97.74**	0.4475
Adamax	97.44	0.5505
Adagrad	97.61	0.4909
Adadelta	97.51	0.4429
SGD	92.16*	3.8450
RMS Prop	97.52	0.4888

Table 5: ReLu Accuracy Score (A) and Standard Deviation (SD)

Overall, we can observe an improvement of 6% over the 91% of the current approach. Nevertheless, using accuracy as the only evaluation metric can lead to misinterpretations since an increase in accuracy might not indicate an increase in predictive ability. To ensure that the increase in accuracy of this methodology corresponds to an increase in performance, we proceeded to compare the two optimizer+activation pairs that had the highest accuracy score (Adagrad+Linear function, and Nadam+ReLu) using precision, recall and f1-score, in order to ensure that the models are capable of generalization.

Opt.	**MAE**	**P**	**R**	**F1**
Adagrad	0.0305	**0.9675**	**0.9829**	**0.9751**
Nadam	0.0325	0.9645	0.9785	0.9714

Table 6: Mean Absolute Error (MAE), Precision (P), Recall (R) and f1-Score (F1) results of the Adagrad and Nadam Optimizers

In Table 6 it can be observed that the Adagrad+Linear function model had better performance than the Nadam+ReLu model in all cases that were evaluated, mainly in recall where the Adagrad+Linear function model obtained 98.29%. In the case of the f1-metric, the Adagrad+Linear function model had better performance by a margin of 0.37%. Nevertheless, both models present a significant improvement over the current state of the art.

5 Discussion

The multiplicative models presented by Mitchell and Lapata (2010) operate using tensor multiplication or word vector cross products. While Bizzoni, Chatzikyriakidis, and Ghanimifard (2017) analyzed the performance of a multiplicative approach, this operation might have created a new vector or representation that lost the lexical information provided by the embeddings, and therefore the performance of the model.

Vector concatenation maintains the sequence and order of the AN pairs that are being feed to the network, but it does not take into account their lexical or semantic relationships. While the dot product of word vectors loses the word order, this measure can interpret the similarity between the word pair that is being analyzed. Moreover, since all the AN pairs follow the same structure, in this context word order or word vector order might be of less importance than the semantic relation between them.

A scalar value reduces the dimensionality of the input from $W \in \mathbb{R}^{300\times600}$ and $b \in \mathbb{R}^{300}$ to a single scalar value $W \in \mathbb{R}^{300\times1}$, thus producing a simpler model with a single feature created based on the word vectors of each component of each AN pair. In our case the metaphoricity vector interprets this scalar value as the lexical-semantic relation between each pair and obtains a representation that determines its metaphoricity.

Regarding the dot product scores of the source *Strength*, we used a t-distributed stochastic neighbor embedding (t-SNE) initialized with principal component analysis (PCA) to reduce the dimensionality of the embeddings from 300 to 2 to visualize the adjectives and their pairing nouns. In Figure 2 it can be observed that nouns ("x"[1]) seem to cluster in the center of the vector space along with both *Strength* adjectives ("triangle").

When performing the same analysis with other sources such as *Depth*[2] (Figure 3), the plot shows that nouns tend to be distributed throughout the vector space in a more sparse manner, which could explain why in the case of *Strength* metaphorical AN pairs tend to have a higher dot product mean.

[1]The gray dots represent all the remaining nouns and adjectives of the vocabulary.

[2]We have chosen Depth because it has a similar number of unique nouns (638) as Strength (625).

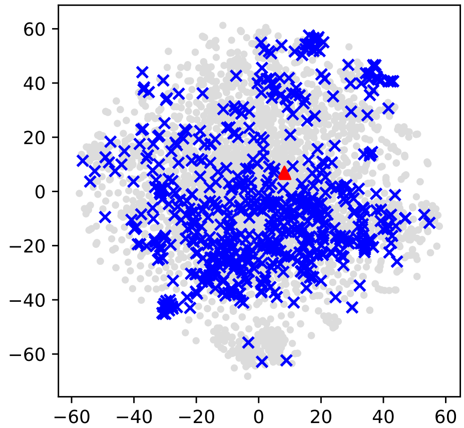

Figure 2: Strength t-SNE-PCA plot

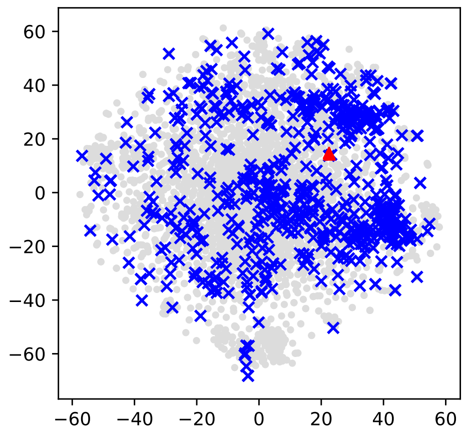

Figure 3: Depth t-SNE-PCA plot

6 Conclusion

In this paper we have presented an approach to AN metaphor detection by implementing a fully connected neural network using pretrained word embeddings. Our multiplicative model consists in computing the dot product between the word vectors of each of the components of the AN pair that is fed to the network. By reducing the dimensionality of the input parameter, this approach introduces a simpler approach to AN metaphor detection while improving the performance of the model.

We evaluated seven optimizers paired with two different activation functions, and in most cases every combination obtained a higher accuracy score in comparison with the current state of the art: an overall of 97% accuracy which represents an improvement of 6% over the 91% reported by Bizzoni, Chatzikyriakidis, and Ghanimifard (2017). To further asses our results we evaluated the top performing models using precision, recall, and f1-score which was not reported in the related works.

Both models obtained 97% in f1-score, and more precisely –after validating the results using 10 K-fold cross validation– the Adagrad+Linear function model obtained 97.51% and the Nadam+ReLu 97.14%. In each instance the only training data where the pre-trained Google News word2vec embeddings, no other features were used during the training process.

Acknowledgements

This research was conducted in the framework of the project *MOMENT: Metaphors of severe mental disorders. Discourse analysis of affected people and mental health professionals*, funded by the Spanish National Research Agency (Agencia Estatal de Investigación, AEI) and the European Regional Development Fund, within the National Programme for Research Aimed at the Challenges of Society. Ref. FFI2017-86969-R (AEI/FEDER, UE)

References

Bizzoni, Y., S. Chatzikyriakidis, and M. 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.

Coll-Florit, M. and S. Climent. 2019. A new methodology for conceptual metaphor detection and formulation in corpora. a case study on a mental health corpus. *SKY Journal of Linguistics*, 32.

Gutierrez, E., E. Shutova, T. Marghetis, and B. Bergen. 2016. Literal and Metaphorical Senses in Compositional Distributional Semantic Models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 183–193, Berlin, Germany. Association for Computational Linguistics.

Lakoff, G. and M. Johnson. 1980. *Metaphors we Live by.* University of Chicago Press, Chicago.

Le, Q. V. and T. Mikolov. 2014. Distributed Representations of Sentences and Documents. *arXiv:1405.4053 [cs]*, May. arXiv: 1405.4053.

Mikolov, T., K. Chen, G. Corrado, and J. Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013b. Distributed Representations of Words and Phrases and their Compositionality. *Proceedings of the 26th International Conference on Neural Information Processing Systems*, 2:9.

Mikolov, T., W. Yih, and G. Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751. Association for Computational Linguistics. event-place: Atlanta, Georgia.

Mitchell, J. and M. Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

Mu, J., H. Yannakoudakis, and E. Shutova. 2019. Learning Outside the Box: Discourse-level Features Improve Metaphor Identification. *arXiv:1904.02246 [cs]*, April. arXiv: 1904.02246.

Pragglejaz Group. 2007. MIP: A method for identifying metaphorically used words in discourse. *Metaphor and Symbol*, 22(1):1–39.

Rosen, Z. 2018. Computationally Constructed Concepts: A Machine Learning Approach to Metaphor Interpretation Using Usage-Based Construction Grammatical Cues. In *Proceedings of the Workshop on Figurative Language Processing*, pages 102–109, New Orleans, Louisiana. Association for Computational Linguistics.

Shutova, E. 2010. Models of Metaphor in NLP. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 688–697.

Shutova, E., 2017. *Annotation of Linguistic and Conceptual Metaphor*, pages 1073–1100. Springer Netherlands, Dordrecht.

Steen, G. J., A. G. Dorst, J. B. Herrmann, A. Kaal, T. Krennmayr, and T. Pasma. 2010. *A Method for Linguistic Metaphor Identification: From MIP to MIPVU*. John Benjamins.

Tsvetkov, Y., L. Boytsov, A. Gershman, E. Nyberg, and C. 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.

Turney, P., Y. Neuman, D. Assaf, and Y. 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., July. Association for Computational Linguistics.

Veale, T., E. Shutova, and B. B. Klebanov. 2016. Metaphor: A Computational Perspective. *Synthesis Lectures on Human Language Technologies*, 9(1):1–160, February.

Wu, C., F. Wu, Y. Chen, S. Wu, Z. Yuan, and Y. Huang. 2018. Neural Metaphor Detecting with CNN-LSTM Model. In *Proceedings of the Workshop on Figurative Language Processing*, pages 110–114, New Orleans, Louisiana. Association for Computational Linguistics.