Metaphor Identification with Paragraph and Word Vectorization: An Attention-Based Neural Approach

Timour Igamberdiev
Department of Linguistics
College of Humanities
Seoul National University
tigamberdiev@snu.ac.kr

Hyopil Shin
Department of Linguistics
College of Humanities
Seoul National University
hpshin@snu.ac.kr

Abstract

The current study investigates approaches to automatic metaphor identification, the computational task of identifying whether a word or phrase in a portion of text is an instance of metaphor. In addition to using the Skip-Gram and Continuous Bag-of-Words algorithms for word-level feature extraction, the Paragraph Vector is utilized for obtaining sentence-level distributional information, being an extension to these two algorithms for blocks of text larger than the word level. With features extracted using the above models, the performance of several different neural network systems are compared against a baseline of logistic regression on the VU Amsterdam Metaphor Corpus, with results showing a significant improvement and high success rates across the different models. This can be seen as strong evidence for the necessity of using state-of-the-art neural network architectures in supervised metaphor identification, being able to pick up on the various latent patterns provided by the vector space model.

1 Introduction

1.1 Background and Overview

The notion of metaphor is very important in language, providing us a glimpse into the cognitive aspects of human linguistic knowledge. Being a fundamental element of our cognition, it allows us to view one particular concept, often more abstract, in terms of another more basic concept, making the former much more accessible to our understanding (Lakoff and Johnson, 1980). Metaphor is ubiquitous in arguably any language, appearing in more subtle forms as in the case of conventional and lexicalized metaphors, as well as very clear cases including novel metaphors, especially within poetry. For example, the following four metaphors, taken from Kövecses (2002), exemplify four different stages of the metaphor SOCIAL ORGANIZATIONS ARE PLANTS:

1. "They had to *prune* the workforce."
2. "Employers *reaped* enormous benefits from cheap foreign labor."
3. "He works for the local *branch* of the bank."
4. "There is a *flourishing* black market in software there."

As mentioned by Croft and Cruse (2004), each of the above metaphors are at different stages in their life-cycle, with the first (1) being the most obvious instance of metaphor, giving a strong sense of the notion of the plant source domain, while the last (4) being nearly unnoticeable to a native speaker, with the word *flourish* having come into English around the year 1300 from the French verb *florir*, having meant 'to blossom' or 'to bloom'.

The distinction between these different types of metaphors is very important for automatic metaphor detection, the computational task of identifying whether a word or phrase in a piece of text is a metaphor or not. Since many recent approaches to computational metaphor processing have used context-based features as input to the classification task (Shutova et al., 2012; Jang et al., 2015; Jang et al., 2016), the degree of success will largely be influenced by which part of the above life cycle the metaphor is in. In other words, depending on the degree to which the metaphor is conventionalized in the language, it is expected to be far more difficult to detect using computational methods, since its domain essentially becomes indistinguishable from the rest of the context. In contrast, a metaphor such as "boiling-hot anger" is far more salient and would be expected to be easily detected by a context-based metaphor classification system, since the source domain (TEMPERATURE) is very distinguishable from the target domain (EMOTION).

In this study, the VU Amsterdam Metaphor Corpus[1] is used as input to a supervised classification task, using the notion of distributional semantic vector spaces and neural network architectures. It is evident that one of the most crucial elements in supervised metaphor classification is the set of features that are used as input to the algorithm (Veale et al., 2016). Throughout research in statistical metaphor processing, a range of features have been looked into, including more concrete information such as semantic roles (Gedigian et al., 2006), domain-types (Dunn, 2013), POS tags, and WordNet super-senses (Hovy et al., 2013), as well as more abstract information, such as features from Latent Dirichlet Allocation (LDA) topic models (Blei et al., 2003) in Beigman Klebanov et al. (2014) and Latent Semantic Analysis (LSA) (Landauer and Dumais, 1997) in Mohler et al. (2014).

The features used in the current set of experiments for metaphor classification are obtained using the log-linear skip-gram and continuous bag-of-words (CBOW) models by Mikolov et al. (2013a), as well as the two algorithms of the Paragraph Vector by Le and Mikolov (2014). Using distributional information from textual data, in this case the VU Amsterdam Metaphor Corpus, latent distributed vector representations are obtained of each sentence from the corpus. If a metaphor appears at least once in the sentence, it is labeled with a '1' as a positive training example and consequently as a '0' if the sentence contains no metaphors. The prepared data is then used as input to various classification algorithms, including a variety of deep neural network models (feedforward neural network and a bidirectional LSTM with attention mechanism).

1.2 Motivation for Methodology

As mentioned above, contextual factors are very relevant to the notion of metaphor. There will inevitably be a semantic contrast between the metaphorically used word and those around it, as can be seen in the above four examples. For instance, 'prune' and 'workforce' in (1) would generally be considered to have relatively distinct meanings, with the former belonging to a domain of discourse related to plants, while the latter to social structures and institutions. In addition, metaphor can be argued to generally appear at semantic units larger than the word level, with an expression such as 'the economy has fallen into a slump', as a whole, representing the mapping from the more abstract notion of economic decline to the more concrete concept of physically falling down.

Hence, the context-based features provided by the skip-gram, CBOW, and Paragraph Vector algorithms are able to accommodate for this need of including contextual information when determining an instance of metaphor, since it is spread out across the vector space for each word and paragraph. In addition, by using a representation at the sentence level, the larger contextual domain is taken into account, with full metaphorical phrases such as the above being represented in the system. The further use of neural network classifiers allows to examine different combinations of the latent semantic qualities depicted in each dimension of the word and sentence embeddings, determining which are the most effective in identifying the presence or absence of a metaphor. Finally, the bidirectional LSTM further accounts for temporal information, which is arguably crucial in processing linguistic data due to the sequential nature of language.

The obtained results show that the bLSTM models with an attention mechanism and word vector input features are the most successful throughout, outperforming the classifiers containing Paragraph Vector features. This can in part be explained by the greater representational capabilities of these more complex models, as well as the richer set of input features, with a particular sentence represented as multiple vectors, as opposed to only one.

[1] Available at http://www.vismet.org/metcor/search/

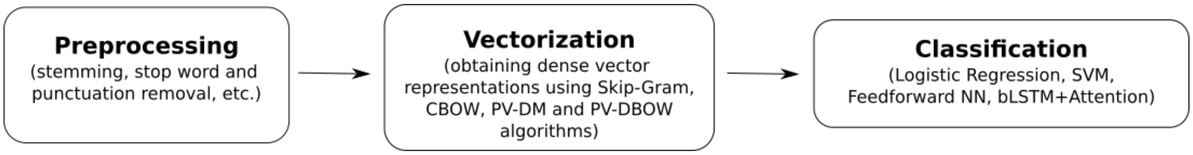
Preprocessing
(stemming, stop word and punctuation removal, etc.)
Vectorization
(obtaining dense vector representations using Skip-Gram, CBOW, PV-DM and PV-DBOW algorithms)
Classification
(Logistic Regression, SVM, Feedforward NN, bLSTM+Attention)

Figure 1: Metaphor Identification Pipeline

2 Data Description

The VU Amsterdam Metaphor Corpus (VUAMC) (Steen et al., 2010) was created from a subset of the British National Corpus, Baby edition, being split into four separate sections: News, Fiction, Academic, and Conversation. Each of the four sections has an average of 47,000 words, with five analysts having manually annotated each word for whether it is a metaphor or non-metaphor, using a procedure derived from the Metaphor Identification Procedure (MIP) (Pragglejaz Group, 2007). The procedure is based on a set of criteria which the annotator must follow in order to mark a particular lexical item as being a metaphor, including whether it has a more basic meaning in other contexts than the one it appears in, with 'basic' referring to being more concrete, less vague, historically older, and so forth.

In the corpus, apart from words counted as 'non-metaphor related', metaphorical words themselves were further subdivided into several different subgroups, collectively considered as 'metaphor-related words' (MRWs). These words were marked as either being 'literal' or 'metaphoric' uses of the metaphor-related word, with the former generally having a metaphor signal present in the context. A third type was the 'implicit' metaphor, which does not have a clear source domain and is based on substitution or ellipsis. Of the three types, the 'metaphoric' use was by far the most common, comprising the majority of the MRW class. In addition, there were several metaphor sub-types that were annotated, including 'PP', being a possible personification, 'Double' metaphors, which contained for instance both a conceptual metaphor and personification, as well as 'WIDLII', which were possible metaphor-related words, but either due to ambiguous context or disagreement among the authors, they were unable to be identified as metaphoric with full certainty. Finally, some words were labeled as metaphor signals, or 'MFlags', which acted as cues for the presence of a metaphor. These words were often, but not always, what are linguistically analyzed as similes (Steen et al., 2010). An instance of the different types of labeled words can be seen in the following set of examples:

1. "On property, he is *blunt*."
   (Clear metaphor, metaphoric use)

2. "At Battersea, Scott could not alter the basic design of the building and he much disliked the '*upturned table*' appearance created by the four corner chimneys."
   (Clear metaphor, literal use, with MFlag)

3. "The information technology revolution has *left* large swaths of rural Britain untouched."
   (PP)

4. "Find a 1980 Toyota Corolla or 1982 Nissan Sunny that has not *succumbed* to rust, and you can buy with confidence."
   (Double)

5. "Auctions certainly speed up the house-buying process. Once the *hammer* has *fallen*, the successful bidder for a house must exchange contracts immediately and pay a deposit."
   (WIDLII)

Within the current study, all 'metaphor-related words' were considered as metaphors for input to the various classifiers, with the rest labeled as non-metaphors. This included words belonging to all the different parts of speech, going beyond a simple analysis of verbs and nouns. The classification task itself was done at the sentence-level, thus any sentence that contains a word labeled as a metaphor was altogether marked as a positive training example. The sentences containing separate metaphor and non-metaphor labels were then used as part of the pipeline in preparing the training and test data for the given binary classification task.

Hyperparameter	Paragraph Vector	Word2Vec
Context Window	10	8
Learning Rate	0.025 (stable)	0.025 (decaying)
Dimension	300	300
Minimum Frequency Count	15	15
Negative Sampling	5	5

Table 1: Hyperparameters of the Paragraph Vector and Word2Vec Algorithms

3 Experiment

The full pipeline of the experiment can be seen in **Figure 1**. First, the extracted corpus is preprocessed, including stemming, stop word and punctuation removal. The corpus is then used to build sentence-level vector representations using the two Paragraph Vector algorithms (PV-DM and PV-DBOW), while the skip-gram and CBOW vectors are made using a separate corpus of the first 1 billion words from Wikipedia. Finally, various classifiers are trained using the obtained embeddings, outlined below. To properly test different hyperparameters, each classifier was tested using stratified 10-fold cross validation, with the mean obtained for accuracy, precision, recall and F-score for any parameters compared.

3.1 Preprocessing

As part of the preprocessing phase, the full corpus data was stemmed using the Snowball Stemmer (Porter, 2001). A short list of stop words was removed, containing common words that arguably do not regularly alternate for metaphor (the, a, an, and, be, is, are, was, were, will). Additional preprocessing steps included changing all letters to lowercase, as well as substituting all number tokens with the '#' sign and removing punctuation. Labels consisted of 8220 positive and 7982 negative examples, being reasonably balanced for both metaphor and non-metaphor classes.

3.2 Vectorization

The Word2Vec model by Mikolov et al. (2013a) consists of two related algorithms, the continuous bag-of-words model (CBOW), in which target words are predicted from input context words within a certain window size, as well as the log-linear skip-gram model, which predicts context words from input target words. For the latter model, this is done by learning a set of parameters θ, given training words $w_1, w_2, w_3, ..., w_T$ within a context window m, such that the following probability is maximized:

$$\prod_{t=1}^{T} \prod_{\substack{-m \leq j \leq m \\ j \neq 0}} P(w_{t+j} \mid w_t; \theta)$$

Converting this to a negative log-likelihood form, the objective becomes to minimize the following:

$$-\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \log P(w_{t+j} \mid w_t)$$

The actual probabilities can be obtained using the softmax function, with $P(w_{t+j} \mid w_t)$ represented as:

$$P(O \mid I) = \frac{exp(u_O^T v_I)}{\sum_{w=1}^{V} exp(u_w^T v_I)}$$

where O is the output word index, I is the input word index, V is the vocabulary size, and v_I and u_O are the input and output vectors with indices of I and O. For purposes of computational efficiency, the model is trained using the notion of Negative Sampling, introduced in Mikolov et al. (2013b).

Both the Distributed Memory Model of Paragraph Vectors (PV-DM), as well as Distributed Bag of Words version of the Paragraph Vector (PV-DBOW) were used for the sentence-level vectorization process. The former is analogous to the above CBOW model for word vector representations, with an additional 'Paragraph Id' included in the input of the shallow neural network architecture, acting as the topic of that paragraph by essentially representing the missing information from that particular context. Similarly, PV-DBOW is analogous to the skip-gram model, with the Paragraph Vector used to predict context words within a text window from that paragraph. Unlike the Word2Vec representations, both PV algorithms produce one vector for each sentence.

Hyperparameter	Feedforward NN	bLSTM with Attention
Input Layer Size	300	89 time steps, 300 units each
Number of Hidden Layers	2	2
Hidden Layer Size	300 * 2	300, 600 (attention)
Output Layer Size	2	1
Dropout	60%	80% (Skip-Gram), 60% (CBOW)
Loss Function	Binary Cross Entropy	Binary Cross Entropy
Optimization Algorithm	Adam	Adam
Epochs	50	5
Activation Function	ReLU, ReLU, Sigmoid	Tanh, Softmax, Sigmoid

Table 2: Hyperparameters of Neural Network Classifiers

3.3 Classification Process

Having created feature representations for all sentences in the corpus with the hyperparameters in **Table 1**, these vectors were then used as input to four different sets of classifiers: Logistic Regression (Cox, 1958) as a baseline, Support Vector Machine (SVM) (Cortes and Vapnik, 1995), Feedforward Neural Network, as well as the Long Short-Term Memory Neural Network (LSTM) (Hochreiter and Schmidhuber, 1997), specifically using a bidirectional implementation (Graves and Schmidhuber, 2005) that contains a word-level attention mechanism, based on the Hierarchical Attention Network of Yang et al. (2016). The first three algorithms were implemented with the PV-DM and PV-DBOW embedding methods, while the fourth algorithm used skip-gram and CBOW vectors.

For the bLSTM, the input layer of the network consisted of 89 time steps, being the length of the longest sentence in the corpus. Prior to input to the bLSTM layer itself, each word went through an embedding layer, in which it was transformed into its corresponding 300-dimensional word vector. As output, a 600-dimensional attention layer was included prior to the final output, allowing the model to *learn* what to focus on from a sequence overall, based on a weighted combination of all the input states to the layer. Specifically, α_{it}, the particular 'importance' of a word vector, is calculated through the activation of the hidden layer for that word, u_{it}, and a learned context vector, u_w, input through a softmax function. A sentence vector is then created by summing these obtained representations:

$$u_{it} = tanh(W_w h_{it} + b_w)$$

$$\alpha_{it} = \frac{exp(u_{it}^{\top} u_w)}{\sum_t exp(u_{it}^{\top} u_w)}$$

$$s_i = \sum_t \alpha_{it} h_{it}$$

Specific parameters of the neural network models can be seen in **Table 2**.

Overall, using a variety of different methods, moving from less complex to more sophisticated architectures, allows for a wide overview of the effectiveness of these different approaches, as well as the degree to which the complexity of the algorithm affects the final results in the metaphor identification task. It is important to investigate the extent to which the distributed embeddings are able to abstract the notion of metaphor from the sentences, in addition to how well the neural network classifiers are able to distinguish and recognize this in those features. Since metaphor can be seen as quite a complex phenomenon, with varying degrees of lexicalization, granularity, as well as a variety of source and target domains, one can hypothesize that classifiers using neural network architectures would be able to discover a lot of subtleties that may not be noticed by more basic models.

4 Results

The results of the classification task for the VUAMC data can be seen in **Table 3** for the PV-DBOW, PV-DM and Word2Vec features, respectively. As expected, overall the neural network classifiers performed better than the baseline of logistic regression. With logistic regression reaching an accuracy of 77.31 and an F-score of about 77.49, it is

Model	Classifier	Accuracy	Precision	Recall	F-Measure
PV-DBOW	LR	77.31 (0.7583)	78.00 (0.8969)	76.98 (0.8443)	77.49 (0.7306)
	SVM	79.08 (0.4882)	80.08 (0.5120)	78.25 (1.291)	79.14 (0.6233)
	FFNN	81.77 (0.5629)	79.66 (1.247)	86.11 (0.9775)	**82.74 (0.3562)**
PV-DM	LR	76.89 (0.5514)	77.37 (0.9093)	77.00 (1.036)	77.17 (0.5253)
	SVM	78.94 (0.8427)	79.43 (0.9937)	78.95 (1.397)	79.18 (0.8716)
	FFNN	78.82 (0.7195)	77.01 (1.057)	83.11 (2.094)	**79.92 (0.8144)**
Skip-Gram	bLSTM	83.73 (1.054)	81.77 (1.845)	87.70 (1.124)	**84.61 (0.8290)**
CBOW	bLSTM	82.40 (1.066)	80.40 (1.354)	86.62 (0.9868)	83.39 (0.9355)

Table 3: Classification Results using Paragraph Vector and Word2Vec Features — Mean (Standard Deviation)

already clear that the PV-DBOW algorithm is providing effective features as input for classification. Apart from the results of the feedforward neural network, for logistic regression and SVM, precision was slightly lower than recall for both the PV-DBOW and PV-DM input features. Especially for PV-DBOW, recall significantly improved with the feedforward neural network, resulting in the highest F-score (82.74) for this classifier out of the tests run on the PV input features. This model also obtained the highest accuracy at 81.77. SVM generally had a performance in-between that of logistic regression and the feedforward neural network, obtaining slightly higher results in accuracy and precision for the PV-DM algorithm.

It is evident that the results of the PV-DM algorithm are overall lower than that of PV-DBOW. With the exception of the SVM reaching about the same recall and F-score for the two algorithms, nearly all other values are lower for PV-DM. Thus, PV-DBOW can definitely be seen as more effective for the current dataset, although this is somewhat unexpected, given Le and Mikolov's original results of PV-DM performing consistently better. However, in comparison with the IMDB dataset used by Le and Mikolov, containing 100,000 movie reviews with several sentences in each, the VUAMC is relatively smaller, with approximately 16,000 sentences, meaning that some difference in results would be expected.

Finally, the attention-based bLSTM with Word2Vec features proved to be the most effective out of all the classifiers, with features from the skip-gram algorithm providing the best results of an F-score of 84.61. It is evident that the addition of the time step dimension, as well as the attention mechanism provide important information for classifying metaphor at the sentence level. The network is able to take into account each word individually, with attention allowing it to learn specifically what to focus on prior to the classification step. This results in far more refined representational capabilities of the model, since more information is taken into account than with simply one vector corresponding to one sentence, as in the case of the PV algorithms.

5 Discussion and Related Work

The above results support the original hypothesis of neural network classifiers being more effective at picking up on various latent aspects of the vector space model provided by the Paragraph Vector. Although previous studies at word-level metaphor identification have utilized the log-linear skip-gram model of Mikolov et al. (2013a), including Shutova et al. (2016) and Bulat et al. (2017), described below, the procedure here is a novel attempt for metaphor identification using neural network classification at the sentence level, comparing the performance of different architectures.

The following is a small sub-sample of previous work that is relevant to the present study, with a wider overview of other methods mentioned in section 1.1. In Shutova et al. (2016), an F-measure of 79 was obtained using a combination of skip-gram features and visual embeddings based on a given phrase. The dataset used, taken from Mohammad et al. (2016), was a set of 647 verb-noun pairs, annotated for metaphoricity. In addition, the same set of experiments was done on another dataset from Tsvetkov et al. (2014), consisting of 1768 annotated adjective-noun pairs, obtaining an F-measure of 75. In Bulat et al. (2017), this same dataset from Tsvetkov et al. (2014) was used to evaluate a system based on representations created from property norms, with an F-measure of 77.

Regarding experiments with a dataset and task very similar to the current study, Dunn (2013) performed a sentence-level metaphor classification on the VUAMC using a variety of features for different systems, such as domain type, semantic similarity and abstractness, obtaining an F-score of 58.

Dunn et al. (2014) is a continuation on the previous study, also looking into sentence-level metaphor detection in the VUAMC. The authors build a language-independent model based on several algorithms that are run in parallel, combining the final result. A major part of the pipeline is the Category Profile Overlap Classifier, in which source and target words are compared in terms of their category information, using data from background corpora. The degree of overlap between these categories is measured among the words, with a low overlap signifying an instance of metaphor. The obtained result on the VUAMC data containing all four registers is an F-score of 70.3.

Finally, a shared task on automatic metaphor detection was performed by several different teams (Leong et al., 2018) with 8 papers prepared in total from the task (Bizzoni and Ghanimifard, 2018; Leong et al., 2018; Mosolova et al., 2018; Mykowiecka et al., 2018; Pramanick et al., 2018; Skurniak et al., 2018; Stemle and Onysko, 2018; Swarnkar and Singh, 2018; Wu et al., 2018). The dataset utilized was the VUAMC, with word-level metaphor detection carried out on two subsets of the data. The first of these included words from all parts of speech, while the second consisted of only classifying verbs.

Two baselines were provided, one consisting of lemmatized unigrams as input features, while the other also containing WordNet semantic classes, in addition to concreteness rating differences between verbs and nouns, as well as adjectives and nouns. Both baselines used logistic regression for the classification process. On the ‘all POS’ task, the baselines reached an F-score of 0.573 and 0.600, respectively. On the ‘verb only’ task, the first baseline achieved an F-score of 0.581, while the second was 0.589. The architectures prepared by the different teams overall consisted of word embeddings with various neural network classification models. The highest F-score achieved was 0.651 for the ‘all POS’ category and 0.672 for the ‘verb only’ category, both by Wu et al. (2018). This system used a combination of Word2Vec features with a CNN and bLSTM model, including information such as POS tags.

Since the VUAMC is considered to be quite a difficult dataset to achieve high results on due to the large amount of lexicalized and conventional metaphors (Veale et al., 2016), high performance is expectedly more difficult to attain in comparison with that of other datasets. The performance of the neural network models in the current study can thus be seen as a significant improvement in the overall research on metaphor identification, increasing the F-score from the previous study on the same task and dataset (Dunn et al., 2014) from 70.3 to 84.61.

The high performance of the bLSTM model shows the importance of including sequential information in the metaphor detection task. This is understandable from the perspective of the necessity of including contextual input for classification, since LSTMs provide information not only about the relation of words in close sequences, but long-term interdependencies as well. Crucially, the attention layer in the network allows to look inside the actual process of identification by the network and see specifically which words are more informative for it in each sentence. The added mechanism of bidirectionality lets the network consider information from both the beginning and end of an input sequence, allowing to mimic the notion of the anticipation of future context by a human interpreter.

Regarding some of the false predictions in the classification process, one explanation stems from varying judgments in what should be considered as positive or negative examples in a manually-annotated body of text. The concept of metaphor can be seen as a non-discrete category, with arguably no delineable boundary between the presence or absence of it in a particular environment, depending rather on various factors outlined above, such as concreteness, presence of a historically older and more basic alternative meaning, and so on. Thus, it is not always clear even for a human annotator, meaning that some discrepancy between the manual annotations and the automatic predictions would

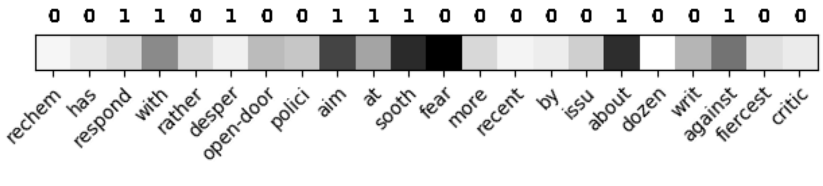

Figure 2: Heat Map from Attention bLSTM

be expected. For example, the sentence "Can't get no birthday money out of him!" was marked as having a metaphor by most of the classifiers, even though it is actually labeled as a non-metaphor in the VUAMC. Although 'out of' can be interpreted here as being a metaphor, with a more concrete alternative meaning denoting a change in the region occupied by a physical trajectory, such as 'out of the room', this was not marked as such by the annotators. With other similar examples present, this is a notion that has to be considered when viewing the classifier's predictions, since varying views on metaphoricity will be reflected in the classification results, as they are in the manual annotation.

Other errors that occurred can be attributed to a lack of context in the sentence for a full evaluation of whether words are used metaphorically or not. An example of such a sentence is "It blows along the valley." Here, without further context, the word 'blows' may be analyzed as non-metaphoric, since it seems to refer to wind. However, the previous sentence provides some essential context that would be necessary for the identification of 'blows' as a metaphor, in which the word actually refers to 'smoke from the factory'. Many such similar examples can further be found in the data, such as "She bought it", in which 'bought' was annotated as a metaphor, but without further context could have either the literal meaning of actually buying something, or the metaphorical meaning of believing somebody.

Figure 2 shows an example heat map of the weights from the attention layer. Blocks labeled as '1' represent words annotated for metaphoricity, while blocks labeled as '0' are non-metaphors. The darker the color is for each block, the more significant the weight for that particular word. It is evident that the model is indeed paying attention to actual instances of metaphor, as with words such as 'aim' and 'about' having much higher weights than that of many other words. In addition, occasionally the model is noticing other elements as well, such as the word 'fear', which is not annotated for metaphoricity in this context. In this way, we are able to look into the actual method of classification by the network, obtaining a window into what is generally considered to be a very enigmatic process. With further investigation into the above patterns of the attention model, it may be possible to get a sense of the general information required for the task of metaphor detection.

6 Conclusion

Using the above methods of classification utilizing neural networks with input features from the PV-DM, PV-DBOW and especially Word2Vec algorithms, it has been demonstrated that a deep learning approach to metaphor identification produces very promising results. Since the system is applicable to new textual data, it is possible to use the trained classifiers on new materials as a practical application for sentence-level metaphor detection, additionally gaining insight into which part of each input sentence is most important in the detection process.

Further work comparing the above classifiers with other neural network architectures would be very interesting, such as the use of Convolutional Neural Networks (CNNs) for classification, as in Kim (2014), as well as other feedforward and recurrent neural models. Furthermore, it would be valuable to see how the above neural models are able to cope with different types of features for metaphor, not only those provided by the Paragraph Vector and Word2Vec algorithms. This would include more concrete features such as domain and semantic information, as well as latent features from other types of models that make sentence-level dense vectors, such as Skip-Thought Vectors (Kiros et al., 2015).

References

Beata Beigman Klebanov, Chee Wee 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, Baltimore, USA.

Yuri Bizzoni and Mehdi Ghanimifard. 2018. In *Bigrams and BiLSTMs: Two Neural Networks for Sequential Metaphor Detection*, pages 91–101, New Orleans, USA.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Luana Bulat, Stephen Clark, and Ekaterina Shutova. 2017. Modelling metaphor with attribute-based semantics. In *Proceedings of 15th Conference of the European Chapter of the Association for Computational Linguistics*, volume 2, pages 523–528, Valencia, Spain. Association for Computational Linguistics (ACL).

Corinna Cortes and Vladimir Vapnik. 1995. Support-Vector Networks. *Machine Learning*, 20:273–297.

David R. Cox. 1958. The Regression Analysis of Binary Sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 20(2):215–242.

William Croft and D. Alan Cruse. 2004. *Cognitive linguistics*. Cambridge University Press, Cambridge.

Jonathan Dunn, Jon Beltran de Heredia, Maura Burke, Lisa Gandy, Sergey Kanareykin, Oren Kapah, Matthew Taylor, Dell Hines, Ophir Frieder, David Grossman, Newton Howard, Moshe Koppel, Scott Morris, Andrew Ortony, and Shlomo Argamon. 2014. Language-independent ensemble approaches to metaphor identification. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 6–12.

Jonathan Dunn. 2013. What metaphor identification systems can tell us about metaphor-in-language. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 1–10, Atlanta, USA.

Matt Gedigian, John Bryant, Srini Narayanan, and Branimir Ciric. 2006. Catching Metaphors. In *Proceedings of the Third Workshop on Scalable Natural Language Understanding*, pages 41–48.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. *Neural Networks*, 18(5):602–610.

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

Dirk Hovy, Shashank Shrivastava, Sujay Kumar Jauhar, Mrinmaya Sachan, Kartik Goyal, Huying Li, Whitney Sanders, and Eduard Hovy. 2013. Identifying Metaphorical Expressions with Tree Kernels. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 52–57, Atlanta, USA.

Hyeju Jang, Seunghwan Moon, Yohan Jo, and Carolyn Penstein Rosé. 2015. Metaphor Detection in Discourse. In *Proceedings of the SIGDIAL 2015 Conference*, pages 384–392, Prague.

Hyeju Jang, Yohan Jo, Qinlan Shen, Michael Miller, Seungwhan Moon, and Carolyn P Rosé. 2016. Metaphor Detection with Topic Transition, Emotion and Cognition in Context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 216–225.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics (ACL).

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-Thought Vectors. In *NIPS*, number 786, pages 1–11.

Zoltan Kövecses. 2002. *Metaphor: A Practical Introduction*. Oxford University Press, Oxford.

George Lakoff and Mark Johnson. 1980. *Methapors We Live By*. University of Chicago Press, Chicago.

Thomas K Landauer and Susan T Dumais. 1997. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge. *Psychological Review*, 104(2):211–240.

Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, Beijing.

Chee Wee Leong, Beata Beigman Klebanov, and Ekaterina Shutova. 2018. In *A Report on the 2018 VUA Metaphor Detection Shared Task*, pages 56–66, New Orleans, USA.

Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, pages 1–12, Scottsdale.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed Representations of Words and Phrases and their Compositionality. In *Advances on Neural Information Processing Systems*, pages 1–9, Lake Tahoe.

Saif M Mohammad, Ekaterina Shutova, and Peter D Turney. 2016. Metaphor as a Medium for Emotion: An Empirical Study. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 23–33, Berlin.

Michael Mohler, Bryan Rink, David Bracewell, and Marc Tomlinson. 2014. A Novel Distributional Approach to Multilingual Conceptual Metaphor Recognition. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, number 2, pages 1752–1763, Dublin, Ireland.

Anna Mosolova, Ivan Bondarenko, and Vadim Fomin. 2018. In *Conditional Random Fields for Metaphor Detection*, pages 121–123, New Orleans, USA.

Agnieszka Mykowiecka, Aleksander Wawer, and Malgorzata Marciniak. 2018. In *Detecting Figurative Word Occurrences Using Recurrent Neural Networks*, pages 124–127, New Orleans, USA.

Martin F. Porter. 2001. Snowball: A language for Stemming Algorithms.

Pragglejaz Group. 2007. MIP: A Method for Identifying Metaphorically Used Words in Discourse. *Metaphor and Symbol*, 22(1):1–39.

Malay Pramanick, Ashim Gupta, and Pabitra Mitra. 2018. In *An LSTM-CRF Based Approach to Token-Level Metaphor Detection*, pages 67–75, New Orleans, USA.

Ekaterina Shutova, Simone Teufel, and Anna Korhonen. 2012. Statistical Metaphor Processing. *Computational Linguistics*, 39(2):301–353.

Ekaterina Shutova, Douwe Kelia, and Jean Maillard. 2016. Black Holes and White Rabbits: Metaphor Identification with Visual Features. In *Proceedings of NAACL-HLT 2016*, pages 160–170. Association for Computational Linguistics (ACL).

Filip Skurniak, Maria Janicka, and Aleksander Wawer. 2018. In *Multi-module Recurrent Neural Networks with Transfer Learning. A Submission for the Metaphor Detection Shared Task*, pages 128–132, New Orleans, USA.

Gerard J. Steen, Aletta G. Dorst, J. Berenike Herrmann, Anna A. Kaal, and Tina Krennmayr. 2010. Metaphor in Usage. *Cognitive Linguistics*, 21(4):765–796.

Egon Stemle and Alexander Onysko. 2018. In *Using Language Learner Data for Metaphor Detection*, pages 133–138, New Orleans, USA.

Krishnkant Swarnkar and Anil Kumar Singh. 2018. In *Di-LSTM Contrast: A Deep Neural Network for Metaphor Detection*, pages 115–120, New Orleans, USA.

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, USA.

Tony Veale, Ekaterina Shutova, and Beata Beigman Klebanov. 2016. *Metaphor: A Computational Perspective*. Morgan & Claypool Publishers, San Francisco.

Chuhan Wu, Fangzhao Wu, Yubo Chen, Sixing Wu, Zhigang Yuan, and Yongfeng Huang. 2018. In *Neural Metaphor Detecting with CNN-LSTM Model*, pages 110–114, New Orleans, USA.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, USA.