Grasping the Finer Point:
A Supervised Similarity Network for Metaphor Detection

Marek Rei♣♠   **Luana Bulat**♣   **Douwe Kiela**◇   **Ekaterina Shutova**♣
♣Computer Laboratory, University of Cambridge, United Kingdom
♠The ALTA Institute, University of Cambridge, United Kingdom
◇Facebook AI Research, New York, USA
{marek.rei,luana.bulat,ekaterina.shutova}@cl.cam.ac.uk, dkiela@fb.com

Abstract

The ubiquity of metaphor in our every-day communication makes it an important problem for natural language understanding. Yet, the majority of metaphor processing systems to date rely on hand-engineered features and there is still no consensus in the field as to which features are optimal for this task. In this paper, we present the first deep learning architecture designed to capture metaphorical composition. Our results demonstrate that it outperforms the existing approaches in the metaphor identification task.

1 Introduction

Metaphor is pervasive in our everyday communication, enriching it with sophisticated imagery and helping us to reconcile our experience in the world with our conceptual system (Lakoff and Johnson, 1980). In the most influential account of metaphor to date, Lakoff and Johnson explain the phenomenon through the presence of systematic metaphorical associations between two distinct concepts or domains. For instance, when we talk about "*curing* juvenile delinquency" or "corruption *transmitting* through the government ranks", we view the general concept of *crime* (the target concept) in terms of the properties of a *disease* (the source concept). Such metaphorical associations are broad generalisations that allow us to project knowledge and inferences across domains; and our metaphorical use of language is a reflection of this process.

Given its ubiquity, metaphorical language poses an important problem for natural language understanding (Cameron, 2003; Shutova and Teufel, 2010). A number of approaches to metaphor processing have thus been proposed, focusing pre-dominantly on classifying linguistic expressions as literal or metaphorical. They experimented with a range of features, including lexical and syntactic information (Hovy et al., 2013; Beigman Klebanov et al., 2016) and higher-level features such as semantic roles (Gedigian et al., 2006), domain types (Dunn, 2013), concreteness (Turney et al., 2011), imageability (Strzalkowski et al., 2013) and WordNet supersenses (Tsvetkov et al., 2014). While reporting promising results, all of these approaches used hand-engineered features and relied on manually-annotated resources to extract them. In order to reduce the reliance on manual annotation, other researchers experimented with sparse distributional features (Shutova et al., 2010; Shutova and Sun, 2013) and dense neural word embeddings (Bracewell et al., 2014; Shutova et al., 2016). Their experiments have demonstrated that corpus-driven lexical representations already encode information about semantic domains needed to learn the patterns of metaphor usage from linguistic data.

We take this intuition a step further and present the first deep learning architecture designed to capture metaphorical composition. Deep learning methods have already been shown successful in many other semantic tasks (e.g. Hermann et al., 2015; Kumar et al., 2015; Zhao et al., 2015), which suggests that designing a specialised neural network architecture for metaphor detection will lead to improved performance. In this paper, we present a novel architecture which (1) models the interaction between the source and target domains in the metaphor via a gating function; (2) specialises word representations for the metaphor identification task via supervised training; (3) quantifies metaphoricity via a weighted similarity function that automatically selects the relevant dimensions of similarity. We experimented with two types of word representations

as inputs to the network: the standard skip-gram word embeddings (Mikolov et al., 2013a) and the cognitively-driven attribute-based vectors (Bulat et al., 2017), as well as a combination thereof.

We evaluate our method in the metaphor identification task, focusing on adjective–noun, verb–subject and verb–direct object constructions where the verbs and adjectives can be used metaphorically. Our results show that our architecture outperforms both a metaphor agnostic deep learning baseline (a basic feed forward network) and the previous corpus-based approaches to metaphor identification. We also investigate the effects of training data on this task, and demonstrate that with a sufficiently large training set our method also outperforms the best existing systems based on hand-coded lexical knowledge.

2 Related Work

The majority of approaches to metaphor processing cast the problem as classification of linguistic expressions as metaphorical or literal. Gedigian et al. (2006) classified verbs related to MO-TION and CURE within the domain of financial discourse. They used the maximum entropy classifier and the verbs' nominal arguments and their FrameNet roles (Fillmore et al., 2003) as features, reporting encouraging results. Dunn (2013) used a logistic regression classifier and high-level properties of concepts extracted from SUMO ontology, including domain types (ABSTRACT, PHYSICAL, SOCIAL, MENTAL) and event status (PROCESS, STATE, OBJECT). Tsvetkov et al. (2014) used random forest classifier and coarse semantic features, such as concreteness, animateness, named entity types and WordNet supersenses. They have shown that the model learned with such coarse semantic features is portable across languages. The work of Hovy et al. (2013) is notable as they focused on compositional rather than categorical features. They trained an SVM with dependency-tree kernels to capture compositional information, using lexical, part-of-speech tag and WordNet supersense representations of sentence trees. Mohler et al. (2013) aimed at modelling conceptual information. They derived semantic signatures of texts as sets of highly-related and interlinked WordNet synsets. The semantic signatures served as features to train a set of classifiers (maximum entropy, decision trees, SVM, random forest) that mapped new metaphors to the semantic signatures of the known ones.

With the aim of reducing the dependence on manually-annotated lexical resources, other research focused on modelling metaphor using corpus-driven information alone. Shutova et al. (2010) pointed out that the metaphorical uses of words constitute a large portion of the dependency features extracted for abstract concepts from corpora. For example, the feature vector for *politics* would contain GAME or MECHA-NISM terms among the frequent features. As a result, distributional clustering of abstract nouns with such features identifies groups of diverse concepts metaphorically associated with the same source domain. Shutova et al. (2010) exploit this property of co-occurrence vectors to identify new metaphorical mappings starting from a set of examples. Shutova and Sun (2013) used hierarchical clustering to derive a network of concepts in which metaphorical associations are learned in an unsupervised way. Do Dinh and Gurevych (2016) investigated metaphors through the task of sequence labelling, detecting metaphor related words in context. Gutiérrez et al. (2016) investigated metaphorical composition in the compositional distributional semantics framework. Their method learns metaphors as linear transformations in a vector space and they demonstrated that it produces superior phrase representations for both metaphorical and literal language, as compared to the traditional "single-sense" compositional distributional model. They then used these representations in the metaphor identification task, achieving promising results.

The more recent approaches of Shutova et al. (2016) and Bulat et al. (2017) used dense skip-gram word embeddings (Mikolov et al., 2013a) instead of the sparse distributional features. Shutova et al. (2016) investigated a set of metaphor identification methods using linguistic and visual features. They learned linguistic and visual representations for both words and phrases, using skip-gram and convolutional neural networks (Kiela and Bottou, 2014) respectively. They then measured the difference between the phrase representation and those of its component words in terms of their cosine similarity, which served as a predictor of metaphoricity. They found basic cosine similarity between the component words in the phrase to be a powerful measure – the neural embeddings of the words were compared with cosine similar-
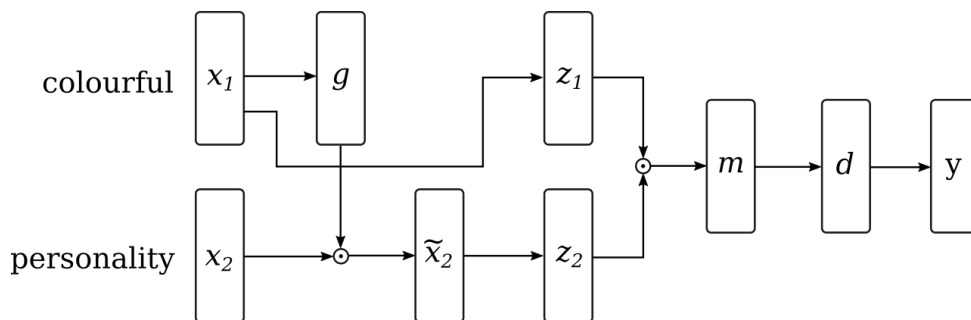
Figure 1: The network architecture for supervised metaphorical phrase classification. The $\odot$ symbol is used to indicate element-wise multiplication.

ity and a threshold was tuned on the development set to distinguish between literal and metaphorical phrases. This approach was their best performing linguistic model, outperformed only by a multi-modal system which included both linguistic and visual features.

Bulat et al. (2017) presented a metaphor identification method that uses representations constructed from human property norms (McRae et al., 2005). They first learn a mapping from the skip-gram embedding vector space to the property norm space using linear regression, which allows them to generate property norm representations for unseen words. The authors then train an SVM classifier to detect metaphors using these representations as input. Bulat et al. (2017) have shown that the cognitively-driven property norms outperform standard skip-gram representations in this task.

3 Supervised Similarity Network

Our method is inspired by the findings of Shutova et al. (2016), who showed that the cosine similarity between neural embeddings of the two words in a phrase is indicative of its metaphoricity. For example, the phrase *'colourful personality'* receives a score:

$$s = cos(x_c, x_p) \tag{1}$$

where x_c is the embedding for *colourful* and x_p is the embedding for *personality*. The combined phrase is classified as being metaphorical based on a threshold, which is optimised on a development dataset. In this paper, we propose several extensions to this general idea, creating a supervised version of the cosine similarity metric which can be optimised on training data to be more suitable for metaphor detection.

3.1 Word Representation Gating

Directly comparing the vector representations of both words treats each of the embeddings as an independent unit. In reality, however, word meanings vary and adapt based on the context. In case of metaphorical language (e.g. "*cure* crime"), the source domain properties of the verb (e.g. *cure*) are projected onto the target domain noun (e.g. *crime*), resulting in the interaction of the two domains in the interpretation of the metaphor.

In order to integrate this idea into the metaphor detection method, we can construct a gating function that modulates the representation of one word based on the other. Given embeddings x_1 and x_2, the gating values are predicted as a non-linear transformation of x_1 and applied to x_2 through element-wise multiplication:

$$g = \sigma(W_g x_1) \tag{2}$$

$$\widetilde{x}_2 = x_2 \odot g \tag{3}$$

where W_g is a weight matrix that is optimised during training, σ is the sigmoid activation function, and $\odot$ represents element-wise multiplication. In an adjective-noun phrase, this architecture allows the network to first look at the adjective, then use its meaning to change the representation of the noun. The sigmoid activation function makes it act as a filter, choosing which information from the original embedding gets through to the rest of the network. While learning a more complex gating function could be beneficial for very large training resources, the filtering approach is more suitable for the annotated metaphor datasets which are relatively small in size.

3.2 Vector Space Mapping

As the next step, we implement position-specific mappings for the word embeddings. The original method uses word embeddings that have been pre-trained using the distributional skip-gram objective (Mikolov et al., 2013a). While this tunes the vectors for predicting context words, there is no reason to believe that the same space is also optimal for the task of metaphor detection. In order to address this shortcoming, we allow the model to learn a mapping from the skip-gram vector space to a new metaphor-specific vector space:

$$z_1 = tanh(W_{z_1} x_1) \tag{4}$$

$$z_2 = tanh(W_{z_2} \widetilde{x}_2) \tag{5}$$

where W_{z_1} and W_{z_2} are weight matrices, z_1 and z_2 are the new position-specific word representations. While the original embeddings x_1 and x_2 are pre-trained on a large unannotated corpus, the transformation process is optimised using annotated metaphor examples, resulting in word representations that are more suitable for this task. Furthermore, the adjectives and nouns use separate mapping weights, which allows the model to better distinguish between the different functionalities of these words. In contrast, the original cosine similarity is not position-specific and would give the same result regardless of the word order.

3.3 Weighted Cosine

If the vectors x_1 and x_2 are normalised to unit length, the cosine similarity between them is equal to their dot product, which in turn is equal to their elementwise multiplication followed by a sum over all elements:

$$cos(x_1, x_2) \propto \sum_i x_{1,i} x_{2,i} \tag{6}$$

This calculation of cosine similarity can be formulated as a small neural network where the two unit-normalised input vectors are directly multiplied together. This is followed by a single output neuron, with all the intermediate weights set to value 1. Such a network would calculate the same sum over the element-wise multiplication, outputting the value of cosine similarity.

Since there is no reason to assume that all the embedding dimensions are equally important when detecting metaphors, we can explore other strategies for weighting the similarity calculation.

Metaphorical	Literal
absorb cost	accommodate guest
attack problem	attack village
attack cancer	blur vision
breathe life	breathe person
design excuse	deflate mattress
deflate economy	digest milk
leak news	land airplane
swallow anger	swim man

Table 1: Annotated verb-direct object and verb-subject pairs from MOH.

Rei and Briscoe (2014) used a fixed formula to calculate weights for different dimensions of cosine similarity and showed that it helped in recovering hyponym relations. We extend this even further and allow the network to use multiple different weighting strategies which are all optimised during training. This is done by first creating a vector m, which is an element-wise multiplication of the two word representations:

$$m_i = z_{1,i} z_{2,i} \tag{7}$$

where m_i is the i-th element of vector m and $z_{1,i}$ is the i-th element of vector z_1. After that, the resulting vector is used as input for a hidden neural layer:

$$d = \gamma(W_d m) \tag{8}$$

where W_d is a weight matrix and γ is an activation function. If the length of d is 1, all the weights in W_d have value 1, and γ is a linear activation, then this formula is equivalent to a regular cosine similarity. However, we use a larger length for d to capture more features, use $tanh$ as the activation function, and optimise the weights of W_d during training, giving the framework more flexibility to customise the model for the task of metaphor detection.

3.4 Prediction and Optimisation

Based on vector d we can output a prediction for the word pair, showing whether it is literal or metaphorical:

$$y = \sigma(W_y d) \tag{9}$$

where W_y is a weight matrix, σ is the logistic activation function, and y is a real-valued prediction with values between 0 and 1.

We optimise the model based on an annotated training dataset, while minimising the following hinge loss function:

$$E = \sum_k q_k \qquad (10)$$

$$q_k = \begin{cases} (\widetilde{y} - y)^2 & \text{if } |\widetilde{y} - y| > 0.4 \\ 0, & \text{otherwise} \end{cases} \qquad (11)$$

where y is the predicted value, $\widetilde{y}$ is the true label, and k iterates over all training examples. Equation 11 optimises the model to minimise the squared error between the predicted and true labels. However, this is only done for training examples where the predicted error is not already close enough to the desired result. The condition $|\widetilde{y} - y| > 0.4$ only updates training examples where the difference from the true label is greater than 0.4. The true labels $\widetilde{y}$ can only take values 0 (literal) or 1 (metaphorical), and the threshold 0.4 is chosen so that datapoints that are on the correct side of the decision boundary by more than 0.1 would be ignored, which helps reduce overfitting and allows the model to focus on the misclassified examples.

The diagram of the complete network can be seen in Figure 1.

4 Word Representations

Following Bulat et al. (2017) we experiment with two types of semantic vectors: skip-gram word embeddings and attribute-based representations.

The word embeddings are 100-dimensional and were trained using the standard log-linear skip-gram model with negative sampling of Mikolov et al. (2013b) on Wikipedia for 3 epochs, using a symmetric window of 5 and 10 negative samples per word-context pair.

We use the 2526-dimensional attribute-based vectors trained by Bulat et al. (2017), following Fagarasan et al. (2015). These representations were induced by using partial least squares regression to learn a cross-modal mapping function between the word embeddings described above and the McRae et al. (2005) property-norm semantic space.

5 Datasets

We evaluate our method using two datasets of phrases manually annotated for metaphoricity.

Metaphorical	Literal
bloody stupidity	bloody nose
deep understanding	cold weather
empty promise	dry skin
green energy	empty can
healthy balance	frosty morning
hot topix	hot chocolate
muddy thinking	gold coin
ripe age	soft leather
sour mood	sour cherry
warm reception	steep hill

Table 2: Annotated adjective–noun pairs from TSV-TEST.

Since these datasets include examples for different senses (both metaphorical and literal) of the same verbs or adjectives, they allow us to test the extent to which our model is able to discriminate between different word senses, as opposed to merely selecting the most frequent class for a given word.

Mohammad et al. dataset (MOH) Mohammad et al. (2016) used WordNet to find verbs that had between three and ten senses and extracted the sentences exemplifying them in the corresponding glosses, yielding a total of 1639 verb uses in sentences. Each of these was annotated for metaphoricity by 10 annotators via the crowd-sourcing platform CrowdFlower[1]. Mohammad et al. selected the verbs that were tagged by at least 70% of the annotators as metaphorical or literal to create their dataset. We extracted verb–direct object and verb–subject relations of the annotated verbs from this dataset, discarding the instances with pronominal or clausal subject or object. This resulted in a dataset of 647 verb–noun pairs (316 metaphorical and 331 literal). Some examples of annotated verb phrases from MOH are presented in Table 1.

Tsvetkov et al. dataset (TSV) Tsvetkov et al. (2014) construct a dataset of adjective–noun pairs annotated for metaphoricity. This is divided into a training set consisting of 884 literal and 884 metaphorical pairs (TSV-TRAIN) and a test set containing 100 literal and 100 metaphorical pairs (TSV-TEST). Table 2 shows a portion of annotated adjective-noun phrases from TSV-TEST. TSV-TRAIN was collected from publicly available metaphor collections on the web and manually

---

[1] www.crowdflower.com

curated by removing duplicates and metaphorical phrases that depend on wider context for their interpretation (e.g. *drowning students*). TSV-TEST was constructed by extracting nouns that co-occur with a list of 1000 frequent adjectives in the TenTen Web Corpus[2] using SketchEngine. The selected adjective-noun pairs were annotated for metaphoricity by 5 annotators with an inter-annotator agreement of $\kappa = 0.76$. Since TSV-TRAIN and TSV-TEST were constructed differently, we follow previous work (Tsvetkov et al., 2014; Shutova et al., 2016; Bulat et al., 2017) and report performance on TSV-TEST. We randomly separated 200 (out of the 1536) examples from the training set to use for development experiments.

6 Experiments and Results

The word representations in our model were initialised with either the 100-dimensional skip-gram embeddings or the 2,526-dimensional attribute vectors (Section 4). These were kept fixed and not updated, which reduces overfitting on the available training examples. For both word representations we use the same embeddings as Bulat et al. (2017), which makes the results directly comparable and shows that the improvements are coming from the novel architecture and are not due to a different embedding initialisation.

The network was optimised using AdaDelta (Zeiler, 2012) for controlling adaptive learning rates. The models were evaluated after each full pass over the training data and training was stopped if the F-score on the development set had not improved for 5 epochs. The transformed embeddings z_1 and z_2 were set to size 300, layer d was set to size 50. The values for these hyperparameters were chosen experimentally using the development dataset. In order to avoid drawing conclusions based on outlier results due to random initialisations, we ran each experiment 25 times with random seeds and present the averaged results in this paper. We implemented the framework using Theano (Al-Rfou et al., 2016) and are making the source code publicly available.[3]

Table 3 contains results of different system configurations on the TSV dataset. The original F-score by Tsvetkov et al. (2014) is still the highest, as they used a range of highly-engineered features that require manual annotation, such as

	Acc	P	R	F1
Tsvetkov et al. (2014)	-	-	-	**85**
Shutova et al. (2016)				
linguistic	-	73	80	76
multimodal	-	67	**96**	79
Bulat et al. (2017)	-	85	71	77
FFN skip-gram	77.6	86.6	65.4	74.4
FFN attribute	76.6	82.0	68.6	74.5
SSN skip-gram	82.2	**91.1**	71.6	80.1
SSN attribute	81.9	86.6	**75.7**	80.6
SSN fusion	**82.9**	90.3	73.8	**81.1**

Table 3: System performance on the Tsvetkov et al. dataset (TSV) in terms of accuracy (Acc), precision (P), recall (R) and F-score (F1).

the lexical abstractness, imageability scores and the relative number of supersenses for each word in the dataset. Our setup is more similar to the linguistic experiments by Shutova et al. (2016), where metaphor detection is performed using pre-trained word embeddings. They also proposed combining the linguistic model with a system using visual word representations and achieved performance improvements. Recently, Bulat et al. (2017) compared different types of embeddings and showed that attribute-based representations can outperform regular skip-gram embeddings.

As an additional baseline, we report the performance on metaphor detection using a basic feed-forward network (FFN). In this configuration, the word embeddings x_1 and x_2 are directly connected to the hidden layer d, skipping all the intermediate network structure. The FFN achieves 74.4% F-score on TSV-TEST, showing that even such a simple model can perform relatively well in a supervised setting. Using attribute vectors instead of skip-gram embeddings gives a slight improvement, especially on the recall metric, which is consistent with the findings by Bulat et al. (2017).

The architecture described in Section 3, which we refer to as a supervised similarity network (SSN), outperforms the baseline and achieves 80.1% F-score using skip-gram embeddings and 80.6% with attribute-based representations. We also created a fusion of these two models where the predictions from both are combined as a weighted average. In this setting, the two networks are trained in tandem and a real-valued weight, which is also optimised during training, is

	Acc	P	R	F1
Shutova et al. (2016)				
linguistic	-	67	76	71
multimodal	-	65	**87**	**75**
FFN skip-gram	71.2	70.4	71.8	70.5
FFN attribute	68.5	66.7	71.0	68.3
SSN skip-gram	**74.8**	**73.6**	**76.1**	**74.2**
SSN attribute	69.7	68.8	69.7	68.8
SSN fusion	70.8	70.1	70.9	69.9

Table 4: System performance on the Mohammad et al. dataset (MOH) in terms of accuracy (Acc), precision (P), recall (R) and F-score (F1).

Input phrase	Gold	Predicted	Score
sunny country	0	0	0.152
sweet treat	0	0	0.358
lost wallet	0	0	0.439
meaningless discussion	0	0	0.150
gentle soldier	0	0	0.175
unforgiving heights	1	1	0.867
easy money	1	1	0.503
blind hope	1	1	0.813
rolling hills	1	1	0.677
educational gap	1	1	0.827
humane treatment	0	1	0.617
democratic candidate	0	1	0.510
rich programmer	0	1	0.514
fishy offer	1	0	0.290
backward area	1	0	0.161
sweet person	1	0	0.332

Table 5: Examples from the Tsvetkov development set, together with the gold label, predicted label, and the predicted score from the best model.

used to combine them together. This configuration achieves 81.1% F-score, indicating that the the skip-gram embeddings and attribute vectors capture somewhat complementary information. Excluding the system by Tsvetkov et al. (2014) which requires hand-annotated features, the proposed similarity network outperforms all the previous systems, even improving over the multimodal system by Shutova et al. (2016) without requiring any visual information. The attribute-based SSN also improves over Bulat et al. (2017) by 5.6% absolute, using the same word representations as input.

Table 4 contains results of different system architectures on the MOH dataset. Shutova et al. (2016) reported 75% F-score on this dataset with a multimodal system, after randomly separating a subset for testing. Since this corpus contains only 647 annotated examples, we instead evaluated the systems using 10-fold cross-validation. The feedforward baseline with skip-gram embeddings returns an F-score that is close to the linguistic configuration of Shutova et al, whereas the best results are achieved by the similarity network with skip-gram embeddings. In this setting, the attribute-based representations did not improve performance – this is expected, as the attribute norms by McRae et al. (2005) are designed for nouns, whereas the MOH dataset is centered on verbs.

Table 5 contains examples from the TSV development set, together with gold annotations and predicted scores. The system confidently detects literal phrases such as *sunny country* and *meaningless discussion*, along with metaphorical phrases such as *unforgiving heights* and *blind hope*. The predicted output disagrees with the annotation on

cases such as *humane treatment* and *rich programmer* – some of these examples could also be argued as being metaphorical, depending on the specific sense of the words. While the system was relatively unsure about the false positives (the scores were close to 0.5), it tended to assign more decisive scores to the false negatives.

7 The Effects of Training Data

Results in Section 6 show that performance on the TSV dataset is higher than the MOH dataset, likely due to the former having more examples available for training. Therefore, we ran an additional experiment to investigate the effect of dataset size on the performance of metaphor detection. Gutiérrez et al. (2016) annotated a dataset of adjective-noun phrases as being literal or metaphorical, and we are able to use this as an additional training resource. While it contains only 23 unique adjectives, the total number of phrases reaches 8,592. We remove any phrases that occur in the development or test data of TSV, then incrementally add the remaining examples to the TSV training data and evaluate on the TSV-TEST.

Figure 2 shows a graph of the system performance, when increasing the training data at intervals of 500. There is a very rapid increase in performance until around 2,000 training points, whereas the existing TSV-TRAIN is limited to 1,336 examples. Providing even more data to the system gives an additional increase that is more gradual. The final performance of the system us-
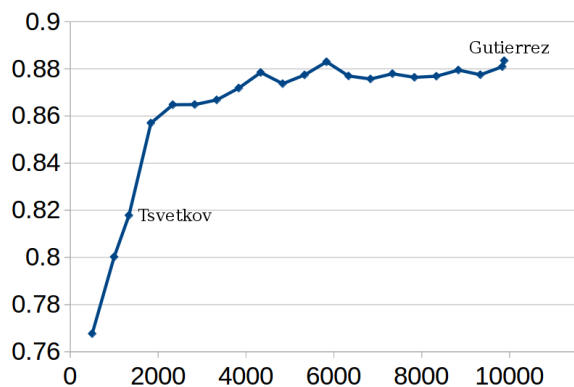
Figure 2: Performance as a function of training set size. The x-axis shows the number of training examples, the y-axis shows F-score on TSV-TEST.

Training data	Acc	P	R	F
Tsvetkov	83.0	88.3	76.3	81.8
Tsvetkov+Gutierrez	**88.7**	**91.6**	**85.4**	**88.3**

Table 6: System performance on the Tsvetkov et al. dataset (TSV), using additional training data.

ing both datasets is 88.3 F-score, which is the highest result reported on the TSV dataset and translates to 36% relative error reduction with respect to the same system trained only on the original dataset.

We report the exact values in Table 6 for the different training sets. The value on the Tsvetkov training data is different from the result in Table 3, which is due to the original attribute embeddings by Bulat et al. (2017) only containing representations for the vocabulary in the TSV dataset. In order to include the data from Gutiérrez et al. (2016), we recreated the attribute vectors for a larger vocabulary, which results in a slightly different baseline performance.

8 Qualitative analysis

The architecture in Section 3 also acts as a semantic composition model, extracting the meaning of the phrase by combining the meanings of its component words. Therefore, we performed a qualitative experiment to investigate: (1) how well do traditional compositional methods capture metaphors, without any fine-tuning; and (2) whether the supervised representations still retain their domain-specific semantic information. For this purpose, we construct three vector spaces and visualise some examples from the TSV training set,
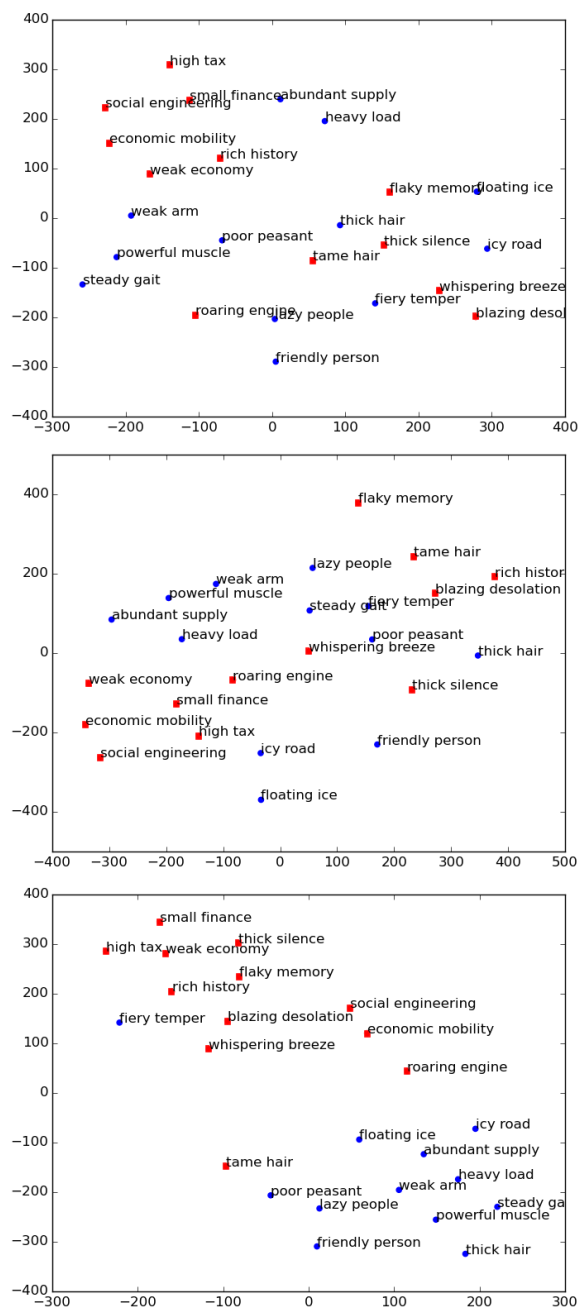

Figure 3: Comparison of metaphorical and literal phrases in different vector spaces. Blue circles indicate literal examples, red squares show metaphorical pairs. Top: additive vector space. Middle: multiplicative vector space. Bottom: vectors from layer m in the similarity network.

using t-SNE (Van Der Maaten and Hinton, 2008).

Figure 3 contains examples for three different composition methods: the additive method simply sums the skip-gram embeddings for both words (top); the multiplicative method multiplies the skip-gram embeddings (middle); the final system uses layer m from the SSN model to represent the

phrases (bottom).

The visualisation shows that the additive and multiplicative models are both comparable when it comes to semantic clustering of the phrases, but metaphorical examples are mixed together with literal clusters. The SSN is optimised for metaphor classification and therefore it produces representations with a very clear boundary for metaphoricity. Interestingly, the graph also reveals a misannotated example in the dataset, since *'fiery temper'* should be labeled as a metaphor. At the same time, this space also retains the general semantic information, as similar phrases with the same label are still positioned close together. Future work could investigate models of multi-task training where metaphor detection is trained together with an unsupervised objective, allowing the system to take better advantage of unlabeled data while still learning to separate metaphors.

9 Conclusion

In this paper, we introduced the first deep learning architecture designed to capture metaphorical composition and evaluated it on a metaphor identification task.

Firstly, we demonstrated that the proposed framework outperforms both a metaphor-agnostic baseline (a feed-forward neural network) as well as previous corpus-driven approaches to metaphor identification. The results showed that it is beneficial to construct a specialised network architecture for metaphor detection, which includes a gating function for capturing the interaction between the source and target domains, word embeddings mapped to a metaphor-specific space, and optimisation using a hinge loss function.

Secondly, our qualitative analysis indicates that our supervised similarity network learns phrase representations with a very clear boundary for metaphoricity, in contrast to traditional compositional methods.

Finally, we show that with a sufficiently large training set our model can also outperform the state-of-the art metaphor identification systems based on hand-coded lexical knowledge.

Acknowledgments

References

Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas, Frédéric Bastien, Justin Bayer, Anatoly Belikov, Alexander Belopolsky, Yoshua Bengio, Arnaud Bergeron, James Bergstra, Valentin Bisson, Josh Bleecher Snyder, Nicolas Bouchard, Nicolas Boulanger-Lewandowski, and Others. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.0:19.

Beata Beigman Klebanov, Chee Wee Leong, E. Dario Gutierrez, Ekaterina Shutova, and Michael Flor. 2016. Semantic classifications for detection of verb metaphors. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 101–106, Berlin, Germany. Association for Computational Linguistics.

David Bracewell, Marc Tomlinson, Michael Mohler, and Bryan Rink. 2014. A tiered approach to the recognition of metaphor. *Computational Linguistics and Intelligent Text Processing*, 8403:403–414.

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 (EACL 2017)*.

Lynne Cameron. 2003. *Metaphor in Educational Discourse*. Continuum, London.

Erik-Lân Do Dinh and Iryna Gurevych. 2016. Token-Level Metaphor Detection using Neural Networks. *Proceedings of the Fourth Workshop on Metaphor in NLP*.

Jonathan Dunn. 2013. Evaluating the premises and results of four metaphor identification systems. In *Proceedings of CICLing'13*, pages 471–486, Samos, Greece.

Luana Fagarasan, Eva Maria Vecchi, and Stephen Clark. 2015. From distributional semantics to feature norms: grounding semantic models in human perceptual data. In *Proceedings of the 11th International Conference on Computational Semantics (IWCS'15)*, pages 52–57, London, UK. Association for Computational Linguistics.

Charles Fillmore, Christopher Johnson, and Miriam Petruck. 2003. Background to FrameNet. *International Journal of Lexicography*, 16(3):235–250.

Matt Gedigian, John Bryant, Srini Narayanan, and Branimir Ciric. 2006. Catching metaphors. In *In Proceedings of the 3rd Workshop on Scalable Natural Language Understanding*, pages 41–48, New York.

E. Darío Gutiérrez, Ekaterina Shutova, Tyler Marghetis, and Benjamin K. Bergen. 2016. Literal and Metaphorical Senses in Compositional

Distributional Semantic Models. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

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

Douwe Kiela and Léon Bottou. 2014. Learning Image Embeddings using Convolutional Neural Networks for Improved Multi-Modal Semantics. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-14)*.

Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. 2015. Ask me anything: Dynamic memory networks for natural language processing. *CoRR, abs/1506.07285*.

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

Ken McRae, George S Cree, Mark S Seidenberg, and Chris McNorgan. 2005. Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods*, 37.

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

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Efficient estimation of word representations in vector space. In *Proceedings of ICLR*, Scottsdale, AZ.

Saif M Mohammad, Ekaterina Shutova, and Peter D Turney. 2016. Metaphor as a medium for emotion: An empirical study. In *Proceedings of *SEM 2016*.

Michael Mohler, David Bracewell, Marc Tomlinson, and David Hinote. 2013. Semantic signatures for example-based linguistic metaphor detection. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 27–35, Atlanta, Georgia.

Marek Rei and Ted Briscoe. 2014. Looking for Hyponyms in Vector Space. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning (CoNLL 2014)*, pages 68–77.

Ekaterina Shutova, Douwe Kiela, and Jean Maillard. 2016. Black Holes and White Rabbits : Metaphor Identification with Visual Features. *Proceedings of NAACL-HLT 2016*.

Ekaterina Shutova and Lin Sun. 2013. Unsupervised metaphor identification using hierarchical graph factorization clustering. In *Proceedings of NAACL 2013*, Atlanta, GA, USA.

Ekaterina Shutova, Lin Sun, and Anna Korhonen. 2010. Metaphor identification using verb and noun clustering. In *Proceedings of Coling 2010*, pages 1002–1010, Beijing, China.

Ekaterina Shutova and Simone Teufel. 2010. Metaphor corpus annotated for source - target domain mappings. In *Proceedings of LREC 2010*, pages 3255–3261, Malta.

Tomek Strzalkowski, George Aaron Broadwell, Sarah Taylor, Laurie Feldman, Samira Shaikh, Ting Liu, Boris Yamrom, Kit Cho, Umit Boz, Ignacio Cases, and Kyle Elliot. 2013. Robust extraction of metaphor from novel data. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 67–76, Atlanta, Georgia.

Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor Detection with Cross-Lingual Model Transfer. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 248–258.

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*, EMNLP '11, pages 680–690, Stroudsburg, PA, USA. Association for Computational Linguistics.

Laurens Van Der Maaten and Geoffrey Hinton. 2008. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9.

Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. *arXiv preprint arXiv:1212.5701*.

Han Zhao, Zhengdong Lu, and Pascal Poupart. 2015. Self-adaptive hierarchical sentence model. *arXiv preprint arXiv:1504.05070*.