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

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 Clement, 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.
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. 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).

54
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 linear 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 adjectives 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).

<table>
<thead>
<tr>
<th>Source</th>
<th>Adjectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature</td>
<td>Cold, heated, icy, warm</td>
</tr>
<tr>
<td>Light</td>
<td>Bright, brilliant, dim</td>
</tr>
<tr>
<td>Texture</td>
<td>Rough, smooth, soft</td>
</tr>
<tr>
<td>Substance</td>
<td>Dense, heavy, solid</td>
</tr>
<tr>
<td>Clarity</td>
<td>Clean, clear, murky</td>
</tr>
<tr>
<td>Taste</td>
<td>Bitter, sour, sweet</td>
</tr>
<tr>
<td>Strength</td>
<td>Strong, weak</td>
</tr>
<tr>
<td>Depth</td>
<td>Deep, shallow</td>
</tr>
</tbody>
</table>

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 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 be-
between them, as shown by Mikolov, Yih, and Zweig (2013), inasmuch as similar words tend to appear near each other within a vector space.

\[ \mathbf{A} \cdot \mathbf{B} = ||\mathbf{A}|| \cdot ||\mathbf{B}|| \cdot \cos \theta \]  

(1)

After computing the dot product for each AN pair, we observed that metaphorical pairs presented a mean result of 0.8548 with a 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](image)

<table>
<thead>
<tr>
<th>Source</th>
<th>Tag</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clarity</td>
<td>Lit.</td>
<td>0.957033</td>
<td>0.705107</td>
</tr>
<tr>
<td></td>
<td>Met.</td>
<td>0.733197</td>
<td>0.544971</td>
</tr>
<tr>
<td>Depth</td>
<td>Lit.</td>
<td>1.564873</td>
<td>0.865550</td>
</tr>
<tr>
<td></td>
<td>Met.</td>
<td>0.778360</td>
<td>0.531733</td>
</tr>
<tr>
<td>Light</td>
<td>Lit.</td>
<td>1.276814</td>
<td>0.859143</td>
</tr>
<tr>
<td></td>
<td>Met.</td>
<td>0.824224</td>
<td>0.647932</td>
</tr>
<tr>
<td>Strength</td>
<td>Lit.</td>
<td>0.628803</td>
<td>0.433746</td>
</tr>
<tr>
<td></td>
<td>Met.</td>
<td>0.799033</td>
<td>0.583103</td>
</tr>
<tr>
<td>Substance</td>
<td>Lit.</td>
<td>1.019069</td>
<td>0.592838</td>
</tr>
<tr>
<td></td>
<td>Met.</td>
<td>0.650521</td>
<td>0.541852</td>
</tr>
<tr>
<td>Taste</td>
<td>Lit.</td>
<td>1.996791</td>
<td>0.884432</td>
</tr>
<tr>
<td></td>
<td>Met.</td>
<td>1.270854</td>
<td>0.887818</td>
</tr>
<tr>
<td>Temperature</td>
<td>Lit.</td>
<td>1.352197</td>
<td>0.938974</td>
</tr>
<tr>
<td></td>
<td>Met.</td>
<td>0.993028</td>
<td>0.770670</td>
</tr>
<tr>
<td>Texture</td>
<td>Lit.</td>
<td>1.299835</td>
<td>0.585611</td>
</tr>
<tr>
<td></td>
<td>Met.</td>
<td>0.699859</td>
<td>0.580966</td>
</tr>
</tbody>
</table>

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.

\[ p = (u, v; \theta) \]  

(2)

\[ p = W^T_{adj} u + W^T_{noun} v + b \]  

(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):

\[ W = \begin{bmatrix} W_{adj} \\ W_{noun} \end{bmatrix} \]  

(4)

\[ p = f_\theta(u, v) = W^T \begin{bmatrix} u \\ v \end{bmatrix} + b \]  

(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 \( u \) over \( v \) (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.

\[ p = f_\theta(u, v) = W_{\text{adj}}^T u \cdot W_{\text{noun}}^T v + b \]  

(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%.

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.

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>A</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adam</td>
<td>97.63</td>
<td>0.4012</td>
</tr>
<tr>
<td>Nadam</td>
<td><strong>97.74</strong></td>
<td>0.4475</td>
</tr>
<tr>
<td>Adamax</td>
<td>97.44</td>
<td>0.5505</td>
</tr>
<tr>
<td>Adagrad</td>
<td>97.61</td>
<td>0.4999</td>
</tr>
<tr>
<td>Adadelta</td>
<td>97.51</td>
<td>0.4429</td>
</tr>
<tr>
<td>SGD</td>
<td>92.16*</td>
<td>3.8450</td>
</tr>
<tr>
<td>RMS Prop</td>
<td>97.52</td>
<td>0.4888</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Opt.</th>
<th>MAE</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adagrad</td>
<td>0.0305</td>
<td><strong>0.9675</strong></td>
<td><strong>0.9829</strong></td>
<td><strong>0.9751</strong></td>
</tr>
<tr>
<td>Nadam</td>
<td>0.0325</td>
<td>0.9645</td>
<td>0.9785</td>
<td>0.9714</td>
</tr>
</tbody>
</table>

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 fed 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 \( W \in \mathbb{R}^{300 \times 600} \) and \( b \in \mathbb{R}^{300} \) to a single scalar value \( W \in \mathbb{R}^{300 \times 1} \), 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 Depth2 (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.

6 Conclusion

In this paper we have presented an approach to AN metaphor detection by implementing a fully connected neural network using pre-trained 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 assess our results we evaluated the

---

1The gray dots represent all the remaining nouns and adjectives of the vocabulary.
2We have chosen Depth because it has a similar number of unique nouns (638) as Strength (625).
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**


