

# Unsupervised Detection of Metaphorical Adjective-Noun Pairs

**Malay Pramanick**

Department of Computer Science  
and Engineering  
IIT Kharagpur  
West Bengal, India - 721302  
malay.pramanick@  
iitkgp.ac.in

**Pabitra Mitra**

Department of Computer Science  
and Engineering  
IIT Kharagpur  
West Bengal, India - 721302  
pabitra@  
cse.iitkgp.ernet.in

## Abstract

Metaphor is a popular figure of speech. Popularity of metaphors calls for their automatic identification and interpretation. Most of the unsupervised methods directed at detection of metaphors use some hand-coded knowledge. We propose an unsupervised framework for metaphor detection that does not require any hand-coded knowledge. We applied clustering on features derived from Adjective-Noun pairs for classifying them into two disjoint classes. We experimented with adjective-noun pairs of a popular dataset annotated for metaphors and obtained an accuracy of 72.87% with k-means clustering algorithm.

## 1 Introduction

Figurative or non-literal elements are ubiquitous in human languages. Usage of non-literal language is popular in day-to-day communications. In this era of Web 2.0, generation of textual data is enormous and thus intractable to be labeled by humans to figure out something from them.

Metaphor is one of the most popular figures of speech. Metaphors are common in online product reviews, blogs, articles and posts in social networking sites. So it has become important for computers to detect metaphors. Interpretation of metaphors comes after their detection in any given text. Also, detection and interpretation of metaphors would definitely help other Natural Language Processing (NLP) tasks like machine translation and summarization.

In 1980, Lakoff and Johnson (1980) proposed Conceptual Metaphor Theory (CMT), in which they claimed that metaphor is not only a property of the language but also a cognitive mechanism that describes our conceptual system. Thus metaphors are devices that transfer the property from

one domain to another unrelated or different domain.

Many supervised as well as unsupervised works have been reported on metaphor detection (Shutova, 2015). Supervised methods require annotated dataset and thus resources are required. Most of the existing unsupervised methods use some hand-coded knowledge, making them hard to scale. Many words can be used metaphorically as well as literally, and words are added to the dictionary on a regular basis. So hand-crafted knowledge about domains cannot be relied upon for a long time, as language is an ever-changing phenomenon necessitating updates of the knowledge base from time to time.

In this paper, we categorically propose an unsupervised framework for metaphor detection without using any hand-coded knowledge, making it robust to scale and adaptive to language change. Using the Adjective-Noun (AN) pairs from the dataset created by Tsvetkov et al. (2014), validations were performed using accuracy as measure and the proposed method demonstrated significant results.

## 2 Related Works

In the recent years, there has been a growing interest in statistical metaphor processing. Many methods, supervised as well as unsupervised, have been proposed for metaphor detection (Shutova, 2015).

Fass (1991) proposed one of the first approaches for metaphor identification and interpretation. The system looked for violated semantic constraints, which are also known as selectional preferences, for identification of metaphors.

TroFi (Trope Finder) (Birke and Sarkar, 2006), is a system that classifies whether a verb is used literally or non-literally, through ‘nearly unsupervised’ techniques. The system is based on statistical word-sense disambiguation techniques (Karov and Edelman, 1998; Stevenson and Wilks, 2003)

and clustering techniques. “TroFi uses sentential context instead of selectional constraint violations or paths in semantic hierarchies” (Birke and Sarkar, 2006).

Wilks et al. (2013) revisited the idea of violation of selectional preferences. To determine whether a sentence contains a metaphor they extracted the subject and direct object for each verb, using the Stanford Parser. After extraction of verbs from the sentence, they checked for preference violations with the help of WordNet (Miller, 1995; Fellbaum, 1998) and VerbNet (Schuler, 2005) and coming across a violation, they marked it as ‘Preference Violation metaphor’. They also considered the ‘conventional metaphors’ and determined them by using the senses in WordNet.

Based on the theory of meaning, Su et al. (2017) presented a metaphor detection technique, considering the difference between the source and target domains in the semantic level rather than the categories of the domains. They extracted subject-object pair by a dependency parser, which they referred to as ‘concept-pair’. They compared the cosine similarity of the concept-pair and from the WordNet, they verified whether the subject was a hypernym or hyponym of the object. When the cosine similarity was below a particular threshold and the ‘concept-pair’ did not have a hypernym-hyponym relation, it was categorized as metaphorical, otherwise literal.

### 3 Motivation and Feature Selection

#### 3.1 Cosine Similarity

Pramanick and Mitra (2017) used cosine similarity to detect metaphors in a supervised way. They showed that cosine similarity of contextually dissimilar words can be used for metaphor detection, which they base on the claim that words have “multiple degrees of similarity”. Their method aims at detecting metaphors in general, so cosine similarity should be helpful in detecting metaphorical Adjective-Noun pairs.

#### 3.2 Abstractness Ratings

According to Köper and im Walde (2017), “abstract words refer to things that can not be seen, heard, felt, smelled, or tasted as opposed to concrete words.” Abstractness of any word is studied by placing the word on a scale ranging between abstract and concrete, known as abstractness ratings. Thus abstractness ratings represent the de-

gree of the abstractness of the thing the word refers to. Abstractness ratings have been shown as a determining factor for metaphor detection (Turney et al., 2011; Dunn, 2013; Tsvetkov et al., 2014; Klebanov et al., 2015; Köper and im Walde, 2016).

#### 3.3 Edit Distance

*Alliteration, assonance and consonance* are figures of speech, in which there is a repetition of letters or sounds. Literary devices are rarely used in isolation, so a way to project the repetitions of letters might help in detection of metaphors, especially if the source of the AN pairs is verse.

To project the repetition of letters, we used edit distance. Given two strings **a** and **b**, the edit distance is the minimum number of edit operations that transforms **a** into **b**. The problem with this representation is that the length of the words varies. So we used the ratio of the edit distance to the length of the word. We considered edit distance from *adjective* to *noun* divided by the length of the *adjective*.

The edit distance is not symmetric. It is not necessarily that  $EditDistance(\mathbf{a}, \mathbf{b}) = EditDistance(\mathbf{b}, \mathbf{a})$ . So we also used the edit distance from *noun* to *adjective*, divided by the length of the *noun*.

#### 3.4 Summary of the Features

The features thus considered are :

1. Abstractness rating of the Adjective
2. Abstractness rating of the Noun
3. Modulus of ( (Abstractness rating of the adjective) - (Abstractness rating of the noun) )
4. Cosine similarity of the Adjective and the Noun
5. Edit distance from the Adjective to the Noun, divided by the length of the Adjective
6. Edit distance from the Noun to the Adjective, divided by the length of the Noun

### 4 Experiments and Results

#### 4.1 Dataset

Tsvetkov et al. (2014) created a large annotated dataset of Adjective-Noun (AN) pairs (henceforth referred to as **TSV** in this paper). The training set *TSV-Train* consists of 884 metaphorical AN pairs and 884 literal AN pairs, and the test set *TSV-Test*

contains 100 metaphorical AN pairs and 100 literal AN pairs. The data was collected by two annotators by using public resources, which was then reduced by at least one additional person “by removing duplicates, weak metaphors and metaphorical phrases (such as *drowning students*) whose interpretation depends on the context”.

Literal	Metaphorical
acute bronchitis	acute ignorance
beaten boxer	beaten path
clouded sky	clouded face
deflated tire	deflated meaning
enormous ship	enormous ego
fragile glass	fragile health
growing plant	growing imbalance
heated oven	heated discussion
shattered glass	shattered dreams
terminal station	terminal poverty
unforgiving soldier	unforgiving heights
velvet jeans	velvet voice
whispering kids	whispering breeze
young girl	young money

Table 1: Annotated AN Pairs from TSV-Train

Literal	Metaphorical
angry protester	angry welt
bald eagle	bald assertion
clear sky	clear explanation
empty can	empty promise
dry skin	dry wit
raw meat	raw emotion
sour cherry	sour mood
white sand	white anger

Table 2: Annotated AN Pairs from TSV-Test

## 4.2 Feature Extraction

We have discussed the features that were used for our experiments and the motivation behind them. Now we discuss how we obtained those features for our experiments. The dataset had some words with accents, which we removed with Unicode (NFKD) normalization during preprocessing, as required for feature extraction.

### 4.2.1 Cosine Similarity

To obtain the vector representation of words, we used the Google word2Vec<sup>1</sup> (Mikolov et al., 2013), an open source tool. We used text corpus from the latest English Wikipedia dump<sup>2</sup> to train the model and obtained word embeddings of dimension 200.

Word vectors were unavailable for some words and most of them contained a hyphen (-). For each of such words, we tried to find its vector by removing the hyphen, still, if the vector was not obtained, we considered the component-wise average of the vector representation of the parts separated by hyphen.

After getting the word vectors for the adjective and the noun, we calculated their cosine similarity, for our experiments.

### 4.2.2 Abstractness Ratings

For our experiments, we used the abstractness ratings proposed by Köper and im Walde (2017). They used “a fully connected feed forward neural network with up to two hidden layers” with word vectors of dimension 300 to obtain the ratings, which have been made public.

We took the abstractness ratings of the adjective and noun and divided each of them by ten (10). The division was performed so as to make the ratings comparable to the cosine similarity, as the abstractness ratings range from 0.0 to 10.0. If the abstractness ratings were not scaled, they could have overshadowed the other features considered.

For the words whose ratings were not available, we tried to obtain the rating by removing the hyphen if present. If the abstractness rating was still not obtained, we tried to obtain the abstractness rating by the taking the average of the abstractness ratings of the parts separated by the hyphen.

### 4.2.3 Edit Distance

With the set of ASCII characters as the alphabet under consideration, the edit operations considered were :

- **Substitution** of a single symbol by another symbol from the alphabet
- **Insertion** of a single symbol from the alphabet
- **Deletion** of a single symbol

<sup>1</sup> Available at <https://code.google.com/archive/p/word2vec/>

<sup>2</sup> Available at <https://dumps.wikimedia.org/enwiki/latest/>

Features	Accuracy
Abstractness Ratings	72.31%
Abstractness Ratings + Cosine Similarity	72.56%
Abstractness Ratings + Cosine Similarity + Edit Distance	72.87%

Table 3: Accuracy of K-Means for Entire Dataset

Features	Training Set	Test Set
Abstractness Ratings	71.21%	82.5%
Abstractness Ratings + Cosine Similarity	71.44%	82.5%
Abstractness Ratings + Cosine Similarity + Edit Distance	71.55%	84%

Table 4: Accuracy of K-Means for Training Data and Test Data

### 4.3 Clustering

K-means was adopted as the clustering algorithm for our experiments. Given a set of  $d$  data points, k-means aims to partition the set into  $k$  ( $k < d$ ) sets. For our experiments, we needed two clusters representing metaphors and literals and we can fix the number of clusters in the k-means clustering algorithm.

First, we ran the k-means algorithm to cluster the entire data provided in the dataset. The algorithm was run with the features described above and without the labels of AN pairs being metaphorical or literal as provided in the dataset. K-means was used to partition the data into two disjoint clusters. Randomly we labeled one of the clusters as metaphorical and the other as literal, and calculated the accuracy. If the calculated accuracy was below 50%, we interchanged the cluster labels and calculated the accuracy. This was done as we had two clusters and we did not know which one was supposed to be metaphorical. The accuracy of the algorithm on the entire data of the dataset is summarized in Table 3.

The dataset comes with divisions of training set and test set. So we ran the k-means clustering algorithm with the training set and obtained the clusters. Similar as above, we measured the accuracy for the training set. With the clusters received after running the clustering algorithm on the training data, we used them to predict the labels (metaphorical or literal) of the test data. As the labels were decided for the clusters of the training data, we used the same labels and report the accuracy in Table 4.

## 5 Discussions

Dependency parsers can be used to extract the nouns along with their adjectival modifiers from running texts to look for Adjective-Noun metaphors or Type-III metaphors as categorized by Krishnakumaran and Zhu (2007). For our experiments, we used TSV, a popular annotated dataset for type-III metaphors.

Turney et al. (2011) used hand-annotated abstractness scores for words to develop their system and reported an accuracy of 0.79 for adjective-noun metaphors but it was rather evaluated on a limited dataset of only 10 adjectives and they had used logistic regression, a supervised method.

Tsvetkov et al. (2014) reported an F-score of 0.85 on the Adjective-Noun classification which is better than the F-score as reported by Shutova et al. (2016). But our method being unsupervised, we cannot compare with their results as they have reported in terms of Precision, Recall and F-score.

## 6 Conclusion

The paper proposes an unsupervised framework for identification of metaphorical adjective-noun word pairs which was evaluated on the large TSV dataset. Cosine similarity and derivatives of abstractness ratings and edit distance were used for clustering.

The proposed framework does not rely on hand-coded knowledge and learns from patterns using machine learning, providing a statistical approach with significant results, which would help as the language changes. The features used in the experiments can also be used for other languages as they are language independent.

## Acknowledgements

We would like to thank Priyanka Sinha and Biswajoy Ghosh for their valuable feedback on the initial draft of this paper. We would also like to thank the anonymous reviewers for their valuable comments and feedback.

## References

- Julia Birke and Anoop Sarkar. 2006. A clustering approach for nearly unsupervised recognition of nonliteral language. In *EACL*.
- 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.
- Dan Fass. 1991. met\*: A method for discriminating metonymy and metaphor by computer. *Computational Linguistics*, 17(1):49–90.
- Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library.
- Yael Karov and Shimon Edelman. 1998. Similarity-based word sense disambiguation. *Computational linguistics*, 24(1):41–59.
- Beata Beigman Klebanov, Chee Wee Leong, and Michael Flor. 2015. Supervised word-level metaphor detection: Experiments with concreteness and reweighting of examples. In *Proceedings of the Third Workshop on Metaphor in NLP*, pages 11–20.
- Maximilian Köper and Sabine Schulte im Walde. 2016. Distinguishing literal and non-literal usage of german particle verbs. In *HLT-NAACL*, pages 353–362.
- Maximilian Köper and Sabine Schulte im Walde. 2017. Improving verb metaphor detection by propagating abstractness to words, phrases and individual senses. *SENSE 2017*, page 24.
- Saisuresh Krishnakumaran and Xiaojin Zhu. 2007. Hunting elusive metaphors using lexical resources. In *Proceedings of the Workshop on Computational approaches to Figurative Language*, pages 13–20. Association for Computational Linguistics.
- George Lakoff and Mark Johnson. 1980. *Metaphors we live by*. University of Chicago press.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- Malay Pramanick and Pabitra Mitra. 2017. A metaphor detection approach using cosine similarity. In *International Conference on Pattern Recognition and Machine Intelligence*, pages 358–364. Springer.
- Karin Kipper Schuler. 2005. Verbnets: A broad-coverage, comprehensive verb lexicon.
- Ekaterina Shutova, Douwe Kiela, and Jean Maillard. 2016. Black holes and white rabbits: Metaphor identification with visual features. In *HLT-NAACL*, pages 160–170.
- Ekaterina Shutova. 2015. Design and evaluation of metaphor processing systems. *Computational Linguistics*, 41(4):579–623.
- Mark Stevenson and Yorick Wilks. 2003. Word sense disambiguation. *The Oxford Handbook of Comp. Linguistics*, pages 249–265.
- Chang Su, Shuman Huang, and Yijiang Chen. 2017. Automatic detection and interpretation of nominal metaphor based on the theory of meaning. *Neurocomputing*, 219:300–311.
- Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer.
- Peter D Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 680–690. Association for Computational Linguistics.
- Yorick Wilks, Adam Dalton, James Allen, and Lucian Galescu. 2013. Automatic metaphor detection using large-scale lexical resources and conventional metaphor extraction. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 36–44.