Metaphor Detection in a Poetry Corpus

Anonymized for blind reviewing

Abstract

Metaphor is indispensable in poetry. It showcases the poet's creativity, and contributes to the overall emotional pertinence of the poem while honing its specific rhetorical impact. Previous work on metaphor detection relies on either rule-based or statistical models, none of them applied to poetry. Our method focuses on metaphor detection in a poetry corpus. It combines rule-based and statistical models (word embeddings) to develop a new classification system. Our system achieved a precision of 0.759 and a recall of 0.804 in identifying metaphor in poetry.

1 Introduction

Metaphor is crucial in the understanding of any literary text. A metaphor deviates from the normal linguistic usage. It intends to create a strong statement that no literal text can accomplish. Metaphor differs from idioms, because one can understand a metaphor even with no prior knowledge. Here are examples of metaphor in poetry:

- The hackles on my neck are fear (Wright, 1958)
- My eyes are caves, chunks of etched rock (Lorde, 2000)

Literary metaphor operates not only in the local context where it appears. It also works in the broader context of the entire work or an author's oeuvre, and in the context of the cultural paradigms associated with a certain specific metaphor field (Ritchie, 2013). Contrary to the standard view, literary metaphor sometimes also maps not only in one direction (from "vehicle" to "tenor") but in two. It thus helps reshape both concepts involved (Ritchie, 2013, p. 189). In other cases, a metaphor interconnects two concepts and so only develops each of them into independent sources of introspective and emotional stimulation (Ritchie, 2013, p. 193).

There is a type of metaphor possibly even more difficult to process automatically: when a whole poem or passage figuratively alludes to an implicit concept. Such is the case, for instance, of Robert Frost's "The Road Not Taken" (Frost, 1962). The poem speaks in its entirety of a consequential choice made in life, without apparently deploying any actual metaphor.

We used a few rule-based methods for metaphor detection as a baseline for our experiments. Turney et al. (2011) proposed the Concrete-Abstract rule: a concrete concept, when used to describe an abstract one, represents a metaphor. A phrase like "Sweet Dreams" is one such example. We use the Abstract-Concrete rule as one of the many features in our model. In experiments, it has in fact proved to be quite useful in the case of poetry as well.

Neuman et al. (2013) propose to categorize metaphor on the basis of POS tag sequences such as Noun-Verb-Noun, Adjective-Noun, etc. We follow the same methodology to extract the set of sentences that can be metaphorical in nature. Our method differs because we use word embeddings pre-trained on the Gigaword corpus (Pennington et al., 2014) to get word vector representations (vector difference and cosine similarity) of possible metaphorical word pairs. Another difference is the addition of two more types of POS sequences, which we have found to be metaphorical in our

Poetry Foundation poetry corpus.[1] We explain the types in section 2.1.

Neuman et al. (2013) describe a statistical model which uses Mutual Information and selectional preferences. The authors suggest using a large-scale corpus to find the concrete nouns most frequently occurring with a specific word. Any word outside this small set denotes a metaphor. Our experiments do not involve finding selectional preference sets directly. Instead, we use word embeddings. We have found the selectional preference sets too limiting. The word span is to be set before the experiments and some sentences exceed that limit, therefore the contextual meaning is lost.

Shutova et al. (2016) introduce a statistical model which detects metaphor just like our method does. Their work, however, is more verb-centered, in that verbs are a seed set for training data. Our work looks more into the possible applications for poetry, not generically. It also focuses more on noun-centered models because, as we have observed, poetry contains more noun-centred than verb-centred metaphor.

Our current work belongs in the same category as the "GraphPoem" project (MARGENTO, 2012; Lou et al., 2015; Tanasescu et al., 2016). The milieu is the computational analysis of poetry, and the goal is the development of tools that can contribute to the academic study of poetry.

2 The Method

2.1 Building the Corpus

We have built our own corpus, because there is no publicly available poetry corpus annotated for metaphor. Annotating poetry line by line can be laborious. We have observed empirically that negative samples are too numerous. To ease this task, we applied Neuman's (2013) approach: consider POS tag sequences to extract potential metaphor. We extracted all sentences from the 12,830 PoFo poems that match these tag sequences.

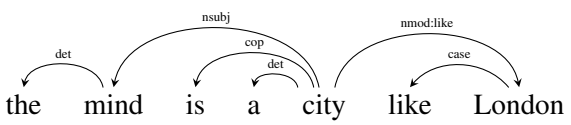
Type I metaphor has a POS tag sequence of Noun-Verb-Noun where the verb is a copula (Neuman et al., 2013). We have extended this to include the tag sequence Noun-Verb-Det-Noun, since we have found that many instances were skipped due to the presence of a determiner. Type II has a tag sequence of Noun-Verb-Noun with a regular, not copula, verb (Neuman et al., 2013). Type III has a tag sequence of Adjective-Noun (Neuman et al., 2013). We also propose two more types that we noticed in our poetry data: Type IV metaphor with a tag sequence of Noun-Verb, and Type V with a tag sequence of Verb-Verb. Here are examples:

- As if the *world were* a *taxi*, you enter it [Type 1] (Koch, 1962)
- I counted the *echoes assembling*, *thumbing* the *midnight* on the piers. [Type 2] (Crane, 2006)
- The moving waters at their *priestlike task* [Type 3] (Keats, 2009)
- The yellow *smoke slipped* by the terrace, made a sudden leap [Type 4] (Eliot, 1915)
- *To die – to sleep* [Type 5] (Shakespeare, 1904)

In this paper, we focus on Type I metaphor. We will work on the remaining four types in the near future. Currently, we are also working on a method independent of POS tag-sequences. It employs a dependency parser (de Marneffe et al., 2006) to give all associations in a sentence. We will use associations such as *nsubj*, *dobj* etc. to filter down to get word pairs that need to be checked for metaphor occurrence. Other irrelevant associations will be discarded. We take this generic approach because we feel that POS sequences may be a little restrictive. Some instances that do not follow the specific POS sequence could be missed.

Identifying head words in a sentence is in itself a challenging task. It is like compressing a phrase to a word pair that may or may not be a metaphor. The POS tag sequence does not always provide an understandable word pair. Sometimes critical words that may be of value are lost. When the nouns highlighted by the POS tagger are not enough to identify the head of a sentence (or a phrase), we use the Stanford NLP Parser (de Marneffe et al., 2006) for identification. As an additional step, we extract all *nsubj* associations from these sentences. If the head word is different from the earlier identified head (suggested by the POS tagger), then the head word is updated.

Here is an example (Schwartz, 1989):



[1] We will abbreviate "Poetry Foundation" to "PoFo" throughout the paper.

2.2 Annotating the Corpus

We extracted around 1500 sentences with the type I metaphor tag sequence, and annotated the first 720. We employed majority voting. First, two independent annotators annotate the 720 sentences without any communication. Then the value of kappa was calculated. Its value came to around 0.39, and agreement to 66.79%. Next, we involved a third annotator who cast a majority vote in case of disagreement. If one of the two annotators agreed to the other's justification, then the disagreement was resolved without the intervention of the third annotator.

While annotating, we encountered several highly ambiguous sentences which required a wider context for assessment. In those rare cases, the annotators were allowed to go back to the poem and judge the metaphor candidate by looking at the context in which it appeared. This was done to avoid discarding a legitimate example for lack of sufficient information. In most cases, however, the sentence alone provided enough information.

All sentences given to the annotators were marked to indicate where the head of the sentence lies. The point was to avoid confusion whenever there were two or more noun phrases. For example:

my eyes are caves , chunks of etched rock @2@ (Lorde, 2000)

The number 2 denotes that the word at location 2, "eyes", is a head word. Therefore the second head would be "caves", because this is a sentence with a Type 1 metaphor tag sequence. Since this is obviously a metaphorical word pair, the annotator would write "y" at the end of the sentence.

The annotators were also allowed to skip a sentence if they could not make up their mind. All in all, a sentence can be labeled as "y" for metaphor, "n" for non-metaphor and "s" for a skipped sentence.

When the annotation process was concluded, we checked for the distribution of classes. Metaphor turned out to be present in 49.8% instances. Non-metaphor accounted for 44.8% and skipped for 5.4%. We had an almost balanced dataset, so we did not need to apply any resampling in our classification. The sentences with skipped annotation were removed from our data. The final dataset contained 680 sentences.

2.3 Rule-based Metaphor Detection

Firstly, we apply rule-based methods to our poetry dataset. We use the Abstract-Concrete (Turney et al., 2011) and Concrete Category Overlap rules (Assaf et al., 2013). The Abstract-Concrete rule needs the hypernym class of each noun; we find that in WordNet (Miller, 1995). We get all hypernyms of head nouns and check for each parent till we reach the hypernym "abstract entity" or "physical entity".

Apart from the above rules, we used a feature based on ConceptNet (Speer and Havasi, 2012). For each noun in our sentence, we extracted the corresponding SurfaceText from ConceptNet. A SurfaceText contain some associations between the specific word and real-world knowledge. For example, "car" gives the following associations:

- "drive" is related to "car"
- You are likely to find "a car" in "the city"

and so on.

The entities are already highlighted in the SurfaceTexts. We parse these associations and extract all the entities. There can be action associations as well:

- "a car" can "crash"
- "a car" can "slow down"

and so on.

These entities and actions are used to establish an overlap in the head nouns of the sentences in the poems. We call this method ConceptNet Overlap. We assign *true* if there is an overlap and *false* otherwise. This is used as one of the features in our rule-based model.

2.4 Statistical-based Metaphor Detection

To capture the distortion of the context that a metaphor causes to a sentence, we compute the vector difference between the vectors for the head words. The underlying idea is as follows: the smaller the difference, the more connected the words would be. Conversely, a significant difference implies disconnected words and hence very likely a metaphor. We render this difference by means of a 100-dimensional vector representation, and we set it as our first statistical feature. Later we test with 200 dimensions as well, to observe the impact on our task.

Experiments	Train	Test	Precision	Recall	F-score
Rules (CA+CCO+CN)	340 PoFo	340 PoFo	0.615	0.507	0.555
PoFo poetry data	340 PoFo	340 PoFo	0.662	0.675	0.669
TroFi data	1771 Tr	1771 Tr	0.797	0.860	0.827
Shutova data	323 Sh	323 Sh	0.747	0.814	0.779
PoFo + TroFi + Shutova	4383 All	487 PoFo	0.759	0.804	0.781

Table 1: Results for the class *metaphor*

To get the word vectors of head words, we used the GloVe vectors pre-trained on the English Gigaword corpus (Pennington et al., 2014). Earlier, we used a custom-trained model based on the British National Corpus (Clear, 1993) but switched to GloVe to test on a larger corpus. Another reason why we tested on two different corpora was to remove any bias that may be perpetuated due to the presence of common-speech metaphor in the corpus. We did not use the available pre-trained word2vec vectors (Mikolov et al., 2013a), because the GloVe vectors had been shown to work better for many lexical-semantic tasks (Pennington et al., 2014).

We did not train word embeddings on the PoFo poems, because the corpus was not large enough for training. Moreover, we needed a corpus that had as few metaphor occurrences as possible, and poetry was obviously not an ideal choice. Training on a poetry corpus would generate word embeddings suited for poems in general, and might miss metaphor instances commonly occurring in poetry. In this task, we were more concerned with the detection of all types of metaphor, not just poetic ones. In effect, distinguishing between common-speech and poetic metaphor has been left for our future work.

We computed the cosine similarity for all word vector pairs, and made it another feature of our model. We also added a feature based on Pointwise Mutual Information in order to measure if a word pair is a collocation:

$$ln\frac{C(x,y).N}{C(x)C(y)}$$

N is the size of the corpus, C(x,y) is the frequency of x and y together, C(x) and C(y) are the frequencies of x and y in corpus, respectively.

3 The Results

We applied our method to the sentences extracted from the 12,830 PoFo poems and annotated manually (see section 2.2). For training data, we used a combination of the datasets such as TroFi (Birke and Sarkar, 2006) and Shutova (Mohammad et al., 2016) with our own poetry dataset. We included other datasets annotated for metaphor, in addition to poetry, in order to increase the training set and thus get better classification predictions. We report all results explicitly for the test set throughout this paper.

Table 1 shows the results for the class *metaphor*. For rule-based experiments, we included Concrete-Abstract, Concrete-Class-Overlap and ConceptNet features (CA, CCO and CN). Training was done on 340 PoFo poem sentences, and testing on the remaining 340 sentences. For PoFo data, training and testing were the same, but with the word vector feature set instead of rules. For the TroFi data, training and testing was done on 1771 instances, each with the same feature set as PoFo. For Shutova's data, training was done on 323 instances and testing on the other 323. Lastly, all the above datasets were aggregated as training data, in order to build a model and to test it on 487 PoFo sentences. Training for this aggregated set was done on 3543 TroFi instances, 647 Shutova instances, and the rest of 193 PoFo instances.

When analyzing the results, one can observe that the TroFi data give the best values overall. Still, when comparing the PoFo results with the aggregate results, we can see that all three metrics have drastically increased when the training data volume increased. Precision on isolated PoFo data is 0.662, whereas on aggregate data it is 0.759. This also establishes that in detecting metaphor in poetry, non-poetry data are as helpful as poetry data.

It can be argued that the recall which we report is not the recall of metaphor throughout the whole poem; instead, it is the recall of the specific POS tag sequence extracted by our algorithm.

	"metaphor"			"literal"		
Classifier	Precision	Recall	F-score	Precision	Recall	F-score
ZeroR	0.565	1.000	0.722	0.000	0.000	0.000
Random Forest	0.741	**0.822**	0.779	**0.731**	0.627	0.675
JRip	0.635	0.745	0.686	0.573	0.443	0.500
J48	0.71	0.615	0.659	0.574	0.675	0.620
KNN	**0.782**	0.756	0.769	0.697	**0.726**	**0.711**
SVM (linear poly.)	0.656	0.742	0.696	0.597	0.495	0.541
SVM (norm. poly.)	0.657	0.767	0.708	0.614	0.481	0.540
SVM (Puk)	0.759	0.804	**0.781**	0.724	0.670	0.696
Naive Bayes	0.663	0.665	0.664	0.564	0.561	0.563
Bayes Net	0.695	0.662	0.678	0.587	0.623	0.604
Adaboost (RF)	0.760	0.713	0.735	0.655	0.708	0.680
Multilayer Perceptron	0.772	0.713	0.741	0.661	0.726	0.692

Table 2: Results for classifiers trained on PoFo+TroFi+Shutova data, and tested on the 487 poetry sentences

Experiments	Train	Test	Precision	Recall	F-score
Rules (CA+CCO+CN)	340 PoFo	340 PoFo	0.462	0.408	0.433
PoFo poetry data	340 PoFo	340 PoFo	0.585	0.570	0.577
TroFi data	1771 Tr	1771 Tr	0.782	0.697	0.737
Shutova data	323 Sh	323 Sh	0.810	0.743	0.775
PoFo + TroFi + Shutova	4383 All	487 PoFo	0.724	0.670	0.696

Table 3: Results for the class *non-metaphor*

There can indeed be sentences that are metaphorical in nature, but are missed due to a different POS tag sequence. We agree with this argument, and are therefore working on a type-independent metaphor identification algorithm to handle those missing cases.

For data preprocessing, we have performed attribute selection by various algorithms, including Pearson's, Infogain and Gain ratio (Yang and Pedersen, 1997). We report the results for the highest accuracy among these algorithms. For classification, we have used the following classifiers: Random Forest, JRip, J48, K-Nearest Neighbor, SVM (Linear Polynomial Kernel), SVM (Normalized Polynomial Kernel), SVM (Pearson Universal Kernel), Naive Bayes, Bayes Net and Multilayer Perceptron.

Table 2 shows a comparison of the results for all classifiers that we tested on the PoFo+TroFi+Shutova data, keeping the training and test set exactly the same. The results are reported on the 487 poetry test data, as noted before. In the case of ZeroR, the classifier just keeps all the instances in the metaphor class, because it is the larger class with 56% of the instances.

For the results in Tables 1 and 3, the SVM classifier (with PUK kernel) was used because it gave the best F-score for the metaphor class (as compared to other classifiers and to SVM with other types of kernels). For attribute selection, we used the Gain ratio evaluator.

Table 3 shows the results for the class *non-metaphor*. It can be noted that – though the precision values of the *metaphor* and *non-metaphor* classes are almost equal – recall of the *non-metaphor* class is lower at 0.670; it is 0.804 for the class *metaphor*. Error analysis showed that these "skipped" cases were mostly archaic words or poetic terms that do not have word vector representations. Still, it is observed that the statistical method scored better than the rule-based method for all metrics.

Table 4 shows a direct comparison between our method (rule-based + statistical) and the methods

Experiments	Method	Precision	Recall	F-score
TroFi (our method)	Rule+Stat	0.797	0.860	**0.827**
TroFi (Birke and Sarkar, 2006)	Active Learning	N/A	N/A	0.649
Shutova (our method)	Rule+Stat	0.747	0.814	**0.779**
Shutova (Shutova et al., 2016)	MIXLATE	0.650	0.870	0.750

Table 4: Results of the direct comparison with related work

of Shutova (2016) and Birke (2006) on their test data (not poetry). Our method performed better than the best-performing method MIXLATE (Shutova et al., 2016) on Mohammad's metaphor data (Mohammad et al., 2016). Our method also performed better than the Active Learning method of Birke and Sarkar (2006) on the TroFi dataset.

We also tested on 200-dimensional word vectors in order to investigate the effect of increasing the number of dimensions from 100 to 200 on accuracy metrics. Results showed that the accuracy dropped by 1%, along with a slight decline in other metrics.

4 Conclusions and Future Work

The preliminary results with Type 1 metaphor encourage us to work more and apply more methods in the future. We are already working on type-independent metaphor identification to increase the recall of our analysis. For rule-based methods, we could work on context overlap methods to remove the ambiguity between various senses that a word may have; this may increase classification accuracy.

For statistical methods, there are many possibilities to look into. Firstly, we are considering analyzing phrase compositionality (Mikolov et al., 2013b) to handle multi-word expressions and phrases better. Since we are identifying metaphor in word pairs rather than the whole sentence, the accuracy of the vector representation for these words is crucial. If a word pair extracted by the algorithm does not represent the whole phrasal meaning, then the classification that follows may obviously prove inaccurate.

Secondly, we are considering deep-learning classifiers such as CNN to further improve precision. Thirdly, we plan to distinguish between poetic and common-speech metaphor. Lastly, we plan to explore ways of quantifying commonalities and hierarchies between metaphor occurrences in order to develop metrics for metaphor quantification. Eventually this metric will be used in the graph rendering, in visualization and in the analysis of poetry corpora.

The recent advances in the field of NLP invite new and more consistent automatic approaches to the study of poetry. We intend to establish that poetry is amenable to computational methods. We also want to demonstrate that the statistical features which this research examines can indeed contribute significantly to the field of digital literary studies, and to academic poetry criticism and poetics in general. A case in point is our observation that non-poetry data are as helpful as poetry data in the task of metaphor detection in poetry.

To the best of our knowledge, this is the first paper on poetic metaphor computational analysis. We hope to see more work in that direction in the future.

References

Dan Assaf, Yair Neuman, Yohai Cohen, Shlomo Argamon, Newton Howard, Mark Last, Ophir Frieder, and Moshe Koppel. 2013. Why "dark thoughts" aren't really dark: A novel algorithm for metaphor identification. In *Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB), 2013 IEEE Symposium on*, pages 60–65. IEEE.

Julia Birke and Anoop Sarkar. 2006. A Clustering Approach for Nearly Unsupervised Recognition of Nonliteral Language. In *Proc. EACL*, pages 329–336.

Jeremy H. Clear. 1993. The British National Corpus. In *The digital word*, pages 163–187. MIT Press.

Hart Crane. 2006. *Complete Poems and Selected Letters*. Library of America.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proc. LREC*, pages 449–454.

Thomas Stearns Eliot. 1915. The Love Song of J. Alfred Prufrock. *Poetry*, 6(3):130–135.

Robert Frost. 1962. The Road Not Taken. In *The Poetry of Robert Frost*. Holt, Rinehart & Winston.

John Keats. 2009. *Bright Star: Love Letters and Poems of John Keats to Fanny Brawne*. Penguin.

Kenneth Koch. 1962. *Thank You, and other Poems*. Grove Press.

Audre Lorde. 2000. *The collected poems of Audre Lorde*. WW Norton & Company.

Andrés Lou, Diana Inkpen, and Chris Tanasescu. 2015. Multilabel Subject-Based Classification of Poetry. In *Proc. FLAIRS*, pages 187–192.

MARGENTO. 2012. *NOMADOSOPHY*. Max Blecher Press.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. 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.

Saif M Mohammad, Ekaterina Shutova, and Peter D. Turney. 2016. Metaphor as a Medium for Emotion: An Empirical Study. In *Proc. *SEM*, pages 23–33.

Yair Neuman, Dan Assaf, Yohai Cohen, Mark Last, Shlomo Argamon, Newton Howard, and Ophir Frieder. 2013. Metaphor Identification in Large Texts Corpora. *PLOS ONE*, 8(4):1–9.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proc. EMNLP*, volume 14, pages 1532–1543.

SL David Ritchie. 2013. Metaphor (Key Topics in Semantics and Pragmatics). *Cambridge university press*, 1(2.1):6.

Delmore Schwartz. 1989. *Last & Lost Poems*, volume 673. New Directions Publishing.

William Shakespeare. 1904. *The Tragedy of Hamlet*. Cambridge University Press.

Ekaterina Shutova, Douwe Kiela, and Jean Maillard. 2016. Black Holes and White Rabbits: Metaphor Identification with Visual Features. In *Proc. 2016 NAACL: HLT*, pages 160–170.

Robert Speer and Catherine Havasi. 2012. Representing General Relational Knowledge in ConceptNet 5. In *Proc. LREC*, pages 3679–3686.

Chris Tanasescu, Bryan Paget, and Diana Inkpen. 2016. Automatic Classification of Poetry by Meter and Rhyme. In *The Twenty-Ninth International Flairs Conference*.

Peter D. Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and Metaphorical Sense Identification through Concrete and Abstract Context. In *Proc. EMNLP*, pages 680–690. Association for Computational Linguistics.

James Wright. 1958. At the Executed Murderer's Grave. *Poetry*, pages 277–279.

Yiming Yang and Jan O. Pedersen. 1997. A Comparative Study on Feature Selection in Text Categorization. In *Proc. ICML*, volume 97, pages 412–420.