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 metaphor detection approaches rely on either rule-based or statistical models, none of them applied to poetry. Our method, focusing on metaphor detection in a poetry corpus, combines rule-based and statistical models (word embeddings) to develop a novel 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 achieve. It is different than an idiom, 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 authors 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, thus helping 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 simulation (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) which in its entirety speaks 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.

Assaf et al. (2013) proposed another algorithm: Concrete Category Overlap (CCO) as an improvement on the Abstract-Concrete rule. It can be applied when both words are concrete and we check for each word's hypernyms categories for overlap. This is used as a feature in our rule-based model.

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 may be metaphorical in nature. Our approach differs as we use word embeddings pretrained 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 that we encountered to be metaphorical in our Poetry Foundation poetry corpus. 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 most frequently occurring concrete nouns with a specific word. Any word outside this small set denotes a metaphor. Our experiments do not directly involve finding selectional preference sets. Instead, we use word embeddings. We find 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. But their work involves more of a verb-centered approach which acts as 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 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

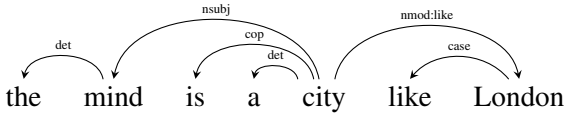
We have built our own corpus, because there is no publicly available poetry corpus annotated for metaphor. Annotating poetry line by line can be laborious. We have observed empirically that negative samples are too numerous. To ease this task, we applied Neuman's (2013) approach: extract potential metaphor based on POS tag sequences. We extracted all sentences from the 12,830 Poetry Foundation 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 extended this to include the tag sequence Noun-Verb-Det-Noun, since we found many cases were skipped due to the presence of a determiner. Type II has a tag sequence of Noun-Verb-Noun with a regular or non-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. Some examples are:

- 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. In our future work, we will work on Types II, III, IV and V. 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. Using associations such as *nsubj*, *dobj* etc., we will filter down to get word pairs that need to be checked for metaphor occurrence. Other irrelevant associations will be discarded. We are working on this generic approach, because we feel that POS sequences may be a bit restrictive; some cases that do not follow the specific POS sequence may 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 and sometimes critical words that may be of value are lost. For these cases, in which the nouns highlighted by the POS tagger are not enough to identify the head of a sentence (or 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 (identified by POS tagger), then the head word is updated.



(Schwartz, 1989)

2.2 Annotating the Corpus

We extracted around 1500 sentences with the type I metaphor tag sequence and annotated the first

720. In annotating, we employed majority voting. First, two independent annotators annotate the 720 sentences without any communication. Then Kappa is calculated. Its value came to around 0.39 and agreement to 66.79%. Later, we involved a third annotator who cast a majority vote in case of disagreement. If one of the two annotators agrees to the other's justification, then the disagreement finishes 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, so that there is no confusion in case there were more than one noun phrase. For example:

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

The number 2 denotes that the word at location 2, i.e., "eyes" is a head word and therefore the second head would be "caves", because this is a Type 1 metaphor tagged sentence. Since this is obviously a metaphorical word pair, the annotator would have to write "y" at the end of the sentence.

Further, the annotators were allowed to skip a sentence, in case they could not make up their mind. Therefore, a sentence can be labeled as "y" for metaphor, "n" for non-metaphor and "s" for skipped sentence.

After annotating, we checked for the distribution of classes. Metaphor turned out to be 49.8%, non-metaphor 44.8% and skipped 5.4%. We have an almost balanced dataset, and therefore we do not need to apply any re-sampling in our classification. The sentences with skipped annotation were removed from our data and therefore the final dataset contained 680 sentences.

2.3 Rule-based Metaphor Detection

Firstly, we use rule-based methods for 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. Therefore we use 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 also used a feature based on ConceptNet (Speer and Havasi, 2012). For each noun in our sentence, we extract the corresponding SurfaceText from ConceptNet. 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 denote *true* if there is an overlap and *false* if not. 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 of the head words. The underlying idea is that 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.

$$\begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ a_n \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \\ . \\ . \\ b_n \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ . \\ . \\ c_n \end{bmatrix}$$

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 class *metaphor*

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 2: Results for class *non-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 bigger corpus. Another reason why we tested on two different corpora was to remove any bias that may perpetuate due to the presence of common speech metaphors in the corpus. We did not use the available pre-trained word2vec vectors (Mikolov et al., 2013a), because the GloVe vectors were shown to work better for many lexical-semantic tasks (Pennington et al., 2014).

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

We computed 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:

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

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

3 The Results

We applied our method on sentences extracted from the 12,830 Poetry Foundation (PoFo) poems and annotated manually. 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 metaphors, in addition to poetry, in order to increase the training set and consequently 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. Training was done on 340 PoFo poem sentences, and testing on the rest of 340 sentences. For PoFo data, the training and test was the same, but with 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 are 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.

On analyzing the results, it can be observed that

		”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 3: Results for different classifiers trained on PoFo+TroFi+Shutova data, and tested on the 487 poetry sentences

Experiments	Method	Precision	Recall	F-score
TroFi (our method)	Rule+Stat	0.797	0.860	**0.827**
TroFi (Birke and Sarkar, 2006)	Active Learning	NA	NA	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

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 was 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 that we report is not the recall of metaphor throughout the whole poem but instead recall of the specific POS tag sequence that is extracted by our algorithm. There can be indeed 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 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 3 shows a comparison of the results from 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 mentioned before. In the case of ZeroR, the classifier just keeps all the instances in the metaphor class as it is the bigger class with 56% of the instances. For the results in Tables 1 and 2, the SVM (with PUK kernel) classifier was used as it gave the best F-score for the metaphor class (as compared to other classifiers and with SVM with other types of kernels). For attribute selection, we used the Gain ratio evaluator.

Table 2 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 words that are archaic or poetic terms that do not have word vec-

tor 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 of Shutova (2016) and Birke (2006) on their test data (non-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 (2006) on the TroFi dataset.

We also tested on 200 dimensional word vectors in order to investigate the impact 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; it may increase classification accuracy.

For statistical methods, there are many possibilities that we are looking 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 the word pair extracted by the algorithm does not represent the whole phrasal meaning, then the later-stage classification may obviously prove inaccurate.

Secondly, we are looking into deploying deep learning classifiers such as CNN to further improve precision. Thirdly, we plan to distinguish between poetic and common-speech metaphors. 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 for graph rendering, visualization and association analysis between poems.

Poetry needs to be computationally researched because the recent advances in NLP have not yet affected this field significantly. We intend to establish that poetry can be studied by computational methods, and that the statistical features this research suggests can indeed be used for academic study of poems. Our observation that to detect metaphor in poetry, non-poetry data is as helpful as poetic one, reinforces this even more. To the best of our knowledge, this is the first paper on the computational analysis of poetic metaphor, and we hope to see more 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*, volume 6, pages 449–454. Genoa.

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 Sstudy. 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*. 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.