METAPHOR OF MIND

The brain is the prisoner of thoughts: a machine-learning assisted quantitative narrative analysis of literary metaphors for use in neurocognitive poetics.
-Arthur M. jacobs & Annette Kinder[9]

ADITI SINGHAL
DEPARTMENT OF COMPUTER SCIENCE
MANAV RACHNA UNIVERSITY
aditisinghal121@gmail.com

ADITI SHARMA
DEPARTMENT OF COMPUTER SCIENCE
MANAV RACHNA UNIVERSITY
aditisharma.5936@gmail.com

RICHA AGGARWAL
DEPARTMENT OF COMPUTER SCIENCE
MANAV RACHNA UNIVERSITY
richaaggarwalstar@gmail.com

1. Introduction

It is not only the pervasiveness of metaphor in human speech but also almost no communication is possible without it, but its enormous potency in expressing and organizing human attitudes which makes it a compelling problem for the social psychologist and even the common people who read it. Also, Metaphor paraphrasing helps to perform quantitative study on the emotions associated with the usage of metaphors.

Besides, it is a very common linguistic phenomenon manifested on average in every third sentence in general-domain text, according to corpus studies [2]. This makes computational processing of metaphors a pressing problem in NLP. There are a number of real-world NLP applications that could benefit from a metaphor processing component, e.g. machine translation [3], opinion mining [4], creative information retrieval [5] and recognizing textual entailment (RTE) [6]. Shutova [3] presents an example from machine translation (MT), where she studied the patterns of metaphor translation from English into Russian by the MT system Google Translate (http://translate.google.com/).

Additionally, even with significant expertise, metaphor detection can be difficult because one metaphor can vary wildly from word to word. Additionally, this task was chosen because it is a fairly straightforward metaphor detection task, and we want to compare conventional machine learning techniques for metaphor detection to more recent deep learning techniques. Specifically, in this report we will compare the use of feature extraction with an SVM(Support Vector Machine), Naive bayes, Random Forest Classifier and Logistic Regression classifier.

Our dataset contains 517*2 phrases, for all the models the output is predicted metaphor.The metrics we chose to compare the methods are accuracy, training and testing and all predictions.

2. Related work

As stated earlier metaphor detection can be done by social psychologists. However, recent times have led to a marked increase in computational methods for metaphor detection.

There are works done in 2013 by Broadwell, in which he argues that metaphors are highly imageable words that do not belong to a discussion topic. To implement this idea, they extend MRC imageability scores to all dictionary words using links among WordNet supersenses (mostly hypernym and hyponym relations) [3].

Strzalkowski et al. (2013) carry out experiments in a specific (government-related) domain for four languages: English, Spanish, Farsi, and Russian. Strzalkowski et al. (2013) explain the algorithm only for English and say that it is the same for Spanish, Farsi, and Russian. Because they heavily rely on WordNet and availability of imageability scores, their approach may not be applicable to low-resource languages [3].

A words sense disambiguation (WSD) is a related problem, where one identifies meanings of polysemous words. The difference is that in the WSD task, we need to select an already existing sense, while for the metaphor detection, the goal is to identify cases of sense borrowing. Studies showed that cross-lingual evidence allows one to achieve a state-of-the-art performance in the WSD task, yet, most cross-lingual WSD methods employ parallel corpora (Navigli, 2009) [3].

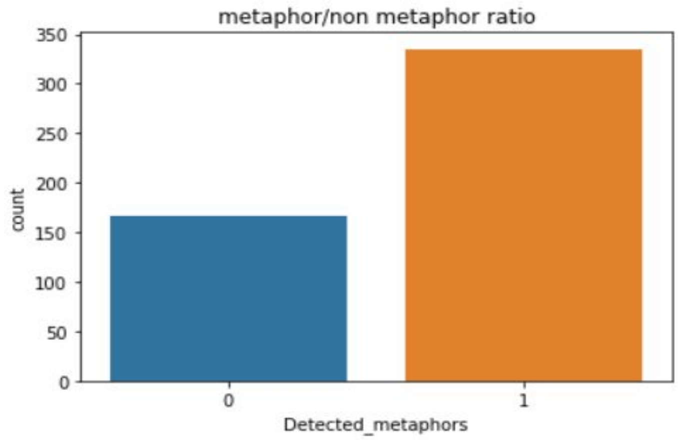
3. Dataset and features

3.1 Obtaining the data

We have used the dataset of metaphorical expressions where a verb is used metaphorically. It is a small dataset of manually annotated metaphorical expressions that contain accurate annotations. It allows us to see how applicable the proposed method is for real world tasks. Our data is largely obtained from the old and modern literature websites, simple nature metaphors, animal idioms, happy and sad sayings. It roughly contains 550 phrases with labels of whether they are metaphors or literals. The expressions in the dataset include e.g. When on earth they fade and perish, The classroom was a zoo (metaphors) and I ate dinner, Oh, how I'd love to go! (literals).

```
In [77]: # counting the value of each class
         df["Detected_metaphors"].value_counts()

Out[77]: 1    335
         0    166
         Name: Detected_metaphors, dtype: int64
```



3.2 Feature Extraction

Since machine learning models do not accept the raw text as input data, we need to convert “Phrases” into vectors of numbers. There are different ways of transforming text into numeric vectors. In this analysis, We’ve applied first the **Bag of Words**, followed by **Tf-Idf** which is a more complex representation. (BoW) is based on the word count statistics.

Bow(w, d)= Number of times word w appears in document d

Tf-Idf stands for term frequency-inverse document frequency, and instead of calculating the counts of each word in each document of the dataset (Bow), it calculates the normalized count where each word count is divided by the number of documents this word appears.

Tf-idf(w, d)= Bow(w, d) * log(Total Number of Documents /(Number of documents in which word w appears)

The dataset was randomly divided into 80/20 train-test splits for the models to first get trained on the words used in metaphors then the testing of the model was done on the testing dataset.

4. Methods

We convert each selected sentence into lowercase and perform tokenization and lemmatization using Python Natural Language Toolkit (NLTK) (http://www.nltk.org).

4.1 Logistic Regression

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. It transforms its output using the logistic sigmoid function to return a probability value.

$$0 \leq h_\theta(x) \leq 1$$

Logistic regression hypothesis expectation

$$f(x) = \frac{1}{1+e^{-(x)}}$$

Formula of a sigmoid function | Image: Analytics India Magazine

4.2 Random Forest

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. We used it because it is a flexible, easy to use that produces, even without hyper-parameter tuning, a great result most of the time. The Random Forest (RF) classifiers are suitable for dealing with the high dimensional noisy data in text classification. An RF model comprises a set of decision trees each of which is trained using random subsets of features.

4.3 Naive Bayes Classifier

The one most suitable for word counts is the **multinomial variant**. Relies on very simple representation of document i.e. Bag of words. From training corpus, extract Vocabulary.

P(x1, x2,..., xn | c)

P(A/B)=P(B/A).P(A)÷P(B) (Bayes Theorem)

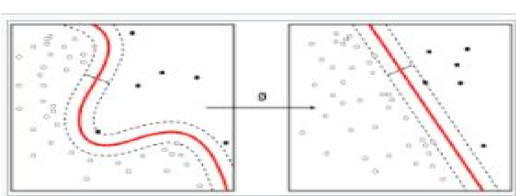
positions ←all word positions in test document

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(x_i \mid c_j)$$

4.4 SVM

Support vector machines is an algorithm that determines the best decision boundary between vectors that belong to a given group (or category) and vectors that do not belong to it. It can be applied to any kind of vector which encodes any kind of data. This means that in order to leverage the power of svm text classification, texts have to be transformed into vectors. Kernel Function, k(x,y).

$$\sum_i \alpha_i k(x_i, x) = \text{constant.}$$



Kernel machine

The theoretical analysis concludes that SVMs acknowledge the particular properties of text:
(a) high dimensional feature spaces,
(b) few irrelevant features (dense concept vectors),
(c) sparse instance vectors.
The experimental results show that SVMs consistently achieve good performance on text categorization tasks, outperforming existing methods substantially and significantly. With their ability to generalize well in high dimensional feature spaces, SVMs eliminate the need for feature selection, making the application of text categorization considerably easier. Furthermore, SVMs do not require any parameter tuning, since they can find good parameter settings automatically. All this makes SVMs a very promising and easy-to-use method for learning text classifiers from examples.

5. Experiments

Model starts learning from a small set of metaphors and then learns patterns of the use of metaphor by means of co-clustering of verbs and nouns. Previously, a lot of work has been done on metaphor in Machine Learning and NLP. We present our own research and argue in favor of adopting data intensive approaches, developing classifiers like SVM, Multinomial Naive Bayes, Logistic Regression and Random Forest in computational literary or poetry analysis, while also highlighting the relevance of such work to DH, NLP, and AI in general. Our models are implemented with the scikit-learn package [19]. The SVM model was trained using the entire training set as one batch. The model for feature combination was then tested on our validation data, and the best was chosen as our best model using the final validation accuracy after training.

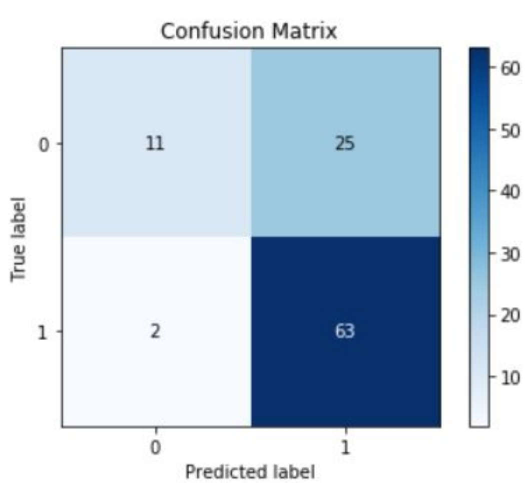
6. Result And Discussion

A workflow model for review processing was established to compare Naïve Bayes, Random Forest, Support Vector Machines, and Logistic Regression classifiers. However, In our comparative work on the text classification with supervised machine learning has concluded that Support Vector Machine is one of the best classifiers, compared to that of Naïve Bayes. Other authors also demonstrated the superiority of Support Vector Machine over random forest, and Naïve Bayes (Dumais et al., 1998). Later, the Support Vector Machine method was chosen by many researchers and became the most popular method for classifying texts. We decided to make a comparison and include a less investigated Logistic Regression classification method, because it is still used in practical tasks as one of the most accurate classification methods.

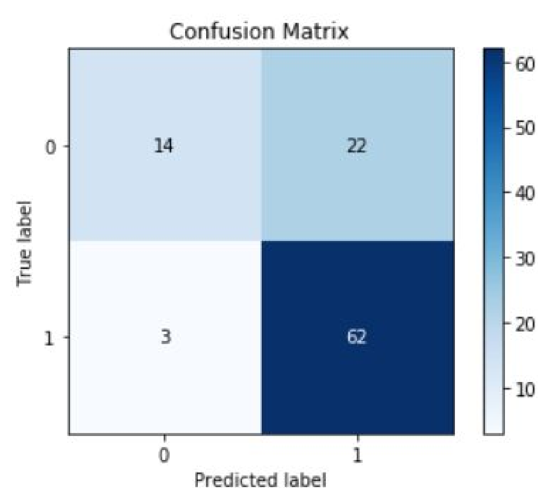
SR.NO.	CLASSIFIER	TESTING ACCURACY	TRAINING ACCURACY
1.	SVM	75.24%	97.75%
2.	Multinomial Naïve Bayes	73.26%	93.75%
3.	Logistic Regression	69.30%	76.75%
4.	Random Forest	52.47%	100.00%

From the results tabulated in the above figure, we immediately notice that our SVM and multinomial Naive Bayes have a severe overfitting issue. The training accuracy is almost 100%, while the test accuracy is much lower. This is in spite of tuning our regularization parameter C across a wide range of values, which indicates that other methods for addressing overfitting should be found and applied, such as using early stopping in the training.

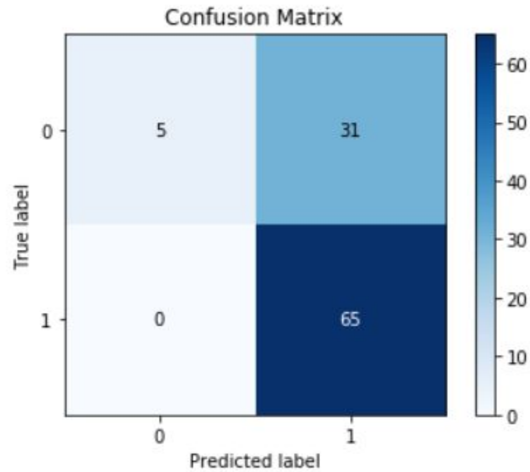
A possible source of the overfitting is the dimension of the feature vectors we use as input; using fewer principal components might reduce the ability of the SVM and Multinomial Naive Bayes to overfit. We can see that compared to even the best SVM results, the Logistic Regression is superior even though accuracy is not very high i.e. maybe due to the Dataset is not consistent but it achieves less noticeable overfitting. Also we can see that Random forest classifiers give very poor results.



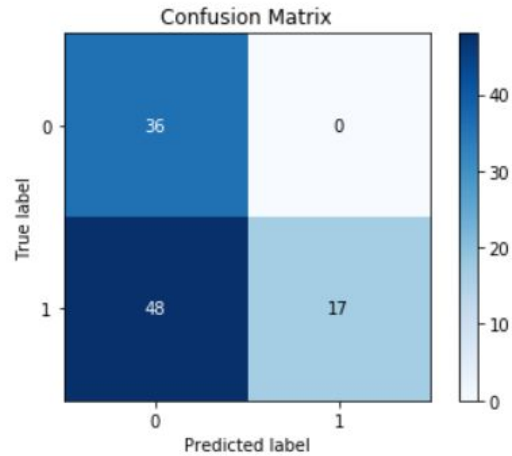
(a) Multinomial Naive Bayes



(b) SVM



(c) Logistic Regression



(d) Random forest Classifier

7. Conclusion And Future Work

The nice thing about text classification is that you have a range of options in terms of what approaches you could use. From unsupervised rules-based approaches to more supervised approaches such as Naive Bayes, SVMs, CRFs and Deep Learning.

We explored the task of metaphor identification using a dataset of 517*2 which includes metaphor as well as phrases and then compare the performance, training and inference time, and ease of implementation for both the classical method of feature extraction with an SVM, Multinomial naive bayes, Random forest and logistic regression.Our best result came from the SVM, with an accuracy of 75.24% in comparison to Random Forest with an accuracy of 52.47%. Therefore, Random Forest is not a good option for this text classification problem.

For future work, we would like to address the overfitting in our Random Forest models. Despite the tuning of the regularization parameter, our Random Forest are overfitting heavily to the training data; this may be mitigated by other regularization techniques such as early stopping. And also we would like to expand the scope of metaphor identification, firstly by using a wide variety of languages and then by including nominal metaphoric relations as well as explore techniques for incorporating contextual features, which can play a key role in identifying certain kinds of metaphors. Second, cross-lingual model transfer can be improved with more careful cross-lingual feature projection.

8. Contributions

The initial dataset, preprocessing and organizing the data was done by Aditi Sharma. We worked together to modify the initial implementations to read in our data in batches in the desired formats. Aditi Sharma and Aditi Singhal collectively worked on iterating over various models to tune hyperparameters, while Richa Aggarwal worked on ingeminate over Random Forest Classifier and Naive Bayes Classifier, and Aditi Singhal managed to explore the literature and research potential additional features for the SVM. The actual implementation of the found features was also done by all three of us.

9. Code

Our code can be found at **https://github.com/aditisinghal651/Metaphor-of-mind**.

10. References

[1] Computationally Constructed Concepts: A Machine Learning Approach to Metaphor Interpretation Using Usage-Based Construction Grammatical Cues. Zachary Rosen Department of Linguistics The University of Colorado, Boulder 295 UCB, Boulder, CO 80309

[2] Shutova E, Teufel S (2010) Metaphor corpus annotated for source - target domain mappings. In: Proceedings of LREC 2010. Malta, 3255–3261.

[3] Shutova E (2011) Computational Approaches to Figurative Language. Ph.D. thesis, University of Cambridge, UK.

[4] Ahmad K (2011) Affective Computing and Sentiment Analysis: Emotion, Metaphor and Terminology. Springer.

[5] Veale T (2011) Creative language retrieval: A robust hybrid of information retrieval and linguistic creativity. In: ACL' 11: 19–24. [Google Scholar]

[6] Agerri R (2008) Metaphor in textual entailment. In: COLING'08 Companion volume – posters and demonstrations. 3–6.

[7] Metaphor Detection with Cross-Lingual Model Transfer Yulia Tsvetkov Leonid Boytsov Anatole Gershman Eric Nyberg Chris Dyer, Language Technologies Institute, Carnegie Mellon University Pittsburgh, PA 15213 USA.

[8] Predicting Human Metaphor Paraphrase Judgments with Deep Neural Networks, Yuri Bizzoni and Shalom Lappin, University of Gothenburg.

[9] The brain is the prisoner of thought: A machine-learning assisted quantitative narrative analysis of literary metaphors for use in Neurocognitive Poetics, Arthur M. Jacobs & Annette Kinder.

[10]Metaphor Interpretation Using Paraphrases Extracted from the Web, Danushka Bollegala ,and Ekaterina Shutova,Neil R. Smalheiser, Editor

[11] Metaphor Detection by Deep Learning and the Place of Poetic Metaphor in Digital Humanities Chris Tanasescu,∗ Vaibhav Kesarwani, Diana Inkpen.

[12] Happy and Sad Sayings - A List of Emotional Phrases

[13] Animal Phrases - Dog and Cat Sayings, Their Meanings and More

[14] Animal Idioms List with Meanings, Examples, & PDFs

[15] https://literaryterms.net/metaphor/

[16] https://blog.reedsy.com/metaphor-examples/

[17] https://literarydevices.net/metaphor/

[18] https://literarydevices.net/10-great-metaphors-from-popular-music/

[19] Pedregosa et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, pp. 2825-2830, 2011.