Literal and Figurative Sense Identification for Phrases using Context

Nazneen Rajani
The University of Texas at Austin
Austin, TX USA
nrajani@cs.utexas.edu

Edaena Salinas
The University of Texas at Austin
Austin, TX USA
edaena@cs.utexas.edu

Abstract

Metaphor is ubiquitous in language, so much so that sometimes it is difficult even for a human judge to detect if a phrase has been used metaphorically. Correct inference about textual entailment requires systems to distinguish the literal and figurative senses of a given phrase. Past work has treated this problem as a classical word sense disambiguation task. In this paper, we take a new approach based on extracting features that are a weighted combination of syntax, semantics, co-occurrence, context and abstractness measure of terms in a text. This approach views detecting figurative sense of phrases as a method for transferring knowledge from a familiar, well-understood, or concrete domain to an unfamiliar, less understood, or more abstract domain. We use the SemEval 2013 phrasal semantics dataset for our experiments and achieve state of-the-art performance on them.

1 Introduction

Idioms are commonly present in a Natural Language. Knowledge of idioms through resources such as dictionaries and knowledge bases has been proven to provide only a limited coverage (Widdows and Dorow, 2005). Hence other methods for identifying idioms have to be explored. One fundamental task is to identify if an idiom or phrase is being used in its figurative or literal sense. Although some phrases rarely occur in their literal sense, there are a number of them which are commonly used in both the literal and figurative senses. As mentioned in (Li and Sporleder, 2010), figurative language detection is an important task in Machine Translation because it helps in preventing mistake of translating a phrase to its literal sense when it was being used figuratively. This task also has applications in Information Retrieval. In (Korkontzelos et al., 2013) illustrate one such application with the following example: *He will go down in history as one of the old school, a true gentleman*. In this case, it would not be correct to retrieve this information for certain queries such as *school*.

In this paper we explored a wide range of features that could be extracted from the context when a phrase appears in a figurative or literal sense. In addition, in (Sporleder and Li, 2009) they determine how well the literal interpretation is linked to the overall cohesive structure of the discourse. We also explored and analyzed the degree of coherence between the phrase and the context in which it appears. We used some external resources like the WordNet to compute relatedness between words. An extension of this idea was to extract lists of topics for the literal and figurative instances of a given phrase and then compute how related a given context was to each of the lists.

We discuss the various features we explored in our approach in Section 2 and we then move on to explain the characteristics of our data and our evaluation metrics and methodology in Section 3.2. Section 4 deals with results and discussion of the performance of our algorithm. We mention related work in Section 5, propose our future work in Section 6 and we conclude in Section 7.

2 Features Identification and Extraction

We now describe our approach to addressing the problem of identifying literal and figurative sense of a given phrase in a given document. We experimented by extracting a variety of features related to word semantics, syntax and co-occurrence. Given below is a list of all features along with their description.

Bag of Words: The (frequency of) occurrence of each word is used as a feature for training a clas-

sifier. In this model, a text (such as a sentence or a document) is represented as an unordered collection of words, disregarding grammar and even word order. The bag of words contributed maximum number of features and also proved to be most effective in our computation.

POS Bag of Words: The Stanford Parser was used for POS tagging the original dataset. We used the POS tagger model that was trained on WSJ sections 0-18 using the left3words architecture. The occurrence of each unique tag was used as a feature for training a classifier. The bow model for POS was chosen to be an initial step into considering the types of POS tags that might appear in a literal context compared to the ones in a figurative context.

Aggregate tf-idf Score: The tf-idf score is the product of the term frequency(tf) and inverse document frequency(idf). It reflects how important a word is to a document in a corpus. This feature comprised of adding up the tf-idf score for each word in the document and then normalizing, as in Equation 1.

$$\text{Aggregate tf-idf(t,d,D)} = \frac{\sum_{t \in T} tf(t,d) \times idf(t,D)}{T} \quad (1)$$

where T is the total number of terms in a document d and D is the total number of documents.

This feature helped in boosting the bag of words features by filtering on stop words and words that occur very commonly in language.

Skip (Gappy) Bigrams: A skip bigram is a pair of tokens in order, possibly with other tokens between them (Goodman, 2000). They are useful for capturing long-distance dependencies between words with sparse data. Consider an example "the cat eats fish." We use two types of skip bigrams. The first type encodes the number of tokens in the gap (e.g., "$the + \_ + \_ + fish$") along with a decay factor so that long gaps are penalized. The second type encodes all gap sizes equally (e.g., "$the + \_\_\_ + eats$" and "$the + \_\_\_ + fish$"). We use gaps containing 1 to 3 tokens for the first type and 1 to 5 tokens for the second type.

Co-occurrence Score: The Co-occurrence score measures the frequency of occurrence of two terms from a text corpus alongside each other in a certain order. In this sense, it can be interpreted as an indicator of semantic proximity. We calculate the co-occurrence score as described by (Turney et al., 2011). We first built a word-context frequency matrix **F** where the rows are the unigrams in training and each phrase is represented with two columns, one marked *left* and one marked *right*. For each phrase, we indicate the frequency with which a unigram occurs to the left of a given phrase and to its right.

We generate a new matrix **X** from the sparse matrix **F** by calculating the Positive Pointwise Mutual Information (PPMI) of each cell in **F** (Turney et al., 2010). The function of PPMI is to emphasize cells in which the frequency is statistically surprising, and hence particularly informative. The matrix **X** was then decomposed with a truncated Singular Value Decomposition (SVD), into product of three matrices $\mathbf{U_k} \sum_{\mathbf{k}} \mathbf{V_k^T}$. The semantic similarity of two terms is given by the cosine of the two corresponding rows in $\mathbf{U_k} \sum_{\mathbf{k}}^{\mathbf{p}}$. The parameter k controls the number of latent factors and the parameter p adjusts the weights of the factors, by raising the corresponding singular values in $\sum_{\mathbf{k}}^{\mathbf{p}}$. We set k to 20 which is a low rank approximation of the co-occurrence matrix and p to 0.5. We did not explore any alternative settings of these parameters. The Co-occurrence score for a training instance is the normalized sum of semantic similarity score for each word in the phrase with every unigram in the instance as shown in Equation 2.

$$\sum_{p \in P} \sum_{u \in U} \frac{sim(p,u)}{|P| \cdot |U|} \quad (2)$$

where p represents a word in the phrase P and u represents a unigram in a instance U. The *sim* is the cosine similarity score.

Concreteness Measure: Concreteness measure is a score that signifies how concrete a given word is. For example, words like *trees*, *walking*, and *red* refer to concrete things, events or properties whereas words such as $economics$, $calculating$, and $disputable$ refer to ideas and concepts that are distant from immediate perception and are hence abstract. We used the MRC Psycholinguistic Database Machine Usable Dictionary (Coltheart, 1981) which contains a list of 4295 words with their concreteness measure between 100 and 700. This feature was the concreteness score sum for the words from the list that occurred in the instances, averaged over the number of words.

Polarity: The Polarity feature signified whether a given instance had positive, negative or neutral po-

larity indicated with a continuous score between -1 and 1. Senti WordNet (Baccianella et al., 2010) provides a list of words along with their positive and negative scores. The aggregate polarity of a word was calculated as $(1 + positiveScore - negativeScore)$. The polarity of a training instance was the normalized sum of the polarities of every word in the document.

Phrase to Context Relatedness: This feature explored how related the words in a phrase are to the context in which it appears. For example, for the phrase *bread and butter*, occurring in literal sense would have words related to food items or eating in the context . As opposed to this, the figurative sense if *bread and butter* may not occur with very closely related words. In order to compute phrase to context relatedness, we used the WordNet, lexical database for English. The first step consisted in obtaining the senses for each word in the context as well as the target phrase. The senses are needed in order to compute the relatedness between words. The senses for the words in the context where computed using the All Words package which assigns a sense to every word in a content by finding the sense of each word that is most related to other words in context based on measures from the WordNet Similarity package(Pedersen and Kolhatkar, 2009). Once each word had a sense, we computed the relatedness between two words using the cosine similarity score, Equation 2.

Topic to Context Relatedness: From training instances, we extracted two lists of topics for each phrase; one for the literal sense and the other for the figurative sense. Topics were extracted using the Maui algorithm used for automatic topic indexing (Medelyan, 2009). For example for the phrase *bread and butter* some of the topics that were obtained were:

> **literal**: tea, food, bread, butter, evening, drink, dish, eating
>
> **figurative**: council, people, great, year, value, oil, west

Once we obtained the topics, we computed how related each word from a context was to each word from the topic list. Given the words from one context in which a phrase occurs, we assigned senses to each word in the two topic lists using cosine similarity measure. We then computed how related each word from the context was to each word from the list of topics. This was done the same way as in the Phrase to Context Relatedness approach.

We decided to apply topic extraction because we considered that the topics obtained from the literal contexts will be more related to each other, and occurrences of a phrase in its literal sense will be more related to the topics from other literal contexts. In contrast, a figurative context will not be as related to the literal topic list as a literal context. We assume that figurative occurrences of a phrase are present in more varied contexts.

Context Words Relatedness: We used the cosine similarity measure between for comparing each word in a given context to every other word in the same context, normalized by the the number of words in the context. This feature was based on the assumption that words in a context have some sort of similarity measure between them.

3 Evaluation

3.1 Data

We experimented on the SemEval 2013 phrasal semantics task (Korkontzelos et al., 2013) datasets. This data was created using a list of English idioms obtained from Wiktionary and entries were manually removed to avoid having phrases which were unlikely to be used in the literal sense. For this list, usage contexts were extracted from the *ukWaC* corpus (Ferraresi et al., 2008) and were labeled as *figurative*, *literal*, *both* or *impossible* using the Crowdsourcing annotation platform CrowdFlower (Korkontzelos et al., 2013). Items with low agreement among workers were not included in the dataset. Table 1 displays more specific characteristics of the datasets.

As shown in table 1, certain instances were classified as being both figurative and literal, for these instances, we replaced *both* with *figuratively* and *literally* and repeating the instance twice. Thus we had altogether 1427 training instances and 595 test instances.

3.2 Evaluation Methodology and Metric

In this section we discuss our method to learn from the training data by extracting features described in Section 2. For features on semantic similarity - Phrase to Context Relatedness and Context Words Relatedness, we experimented with finding a threshold value on the features for classifying instances as literal or figurative. This was done

Table 1: Characteristics of the Dataset

Dataset	# Items	# Items per phrase	# Literal	# Figurative	# Both
train	1,424	68-188	702	719	3
test	594	17-47	294	299	1

using the Threshold Selector meta-classifier from WEKA tool for learning algorithms (Hall et al., 2009). Since most of our features did not have a definite range, we decided to use LIBLINEAR's (Fan et al., 2008) $L2$ regularized Logistic Regression $(L2LR)$.

In accordance with SemEval's specifications, "micro-average" is the accuracy over all instances in test and "macro-average" is the accuracy averaged over each phrase in test. For "macro-average" we trained $L2LR$ for each phrase independently by initializing the weight parameter to the figurative and literal class priors respectively for each run of the experiment. The cost parameter was set to $C = 1$ while calculating the macro-average and was tuned using the development set while calculating the micro-average. The $L2LR$ parameters have been discussed further in Section 4.

$$Accuracy = \frac{\text{\# instances classified correctly}}{\text{\# instances}} \quad (3)$$

In order to evaluate the performance of our system on the test data, we calculated $F1$ score which is the harmonic mean of $precision$ and $recall$ for each class, literal and figurative as defined by (Goutte and Gaussier, 2005). We also evaluate the overall accuracy using Equation 3.

4 Results and Discussion

In this section we present our results and compare them to the results of teams that participated in this SemEval 2013 task. This sections also deals with the discussion and analysis of the results obtained.

4.1 Results

As discussed earlier in Section 3.2, we used $L2$ regularized Logistic Regression for classification. The cost parameter C for calculating the "micro-average" was tuned using the development set. Figure 1 shows the plot of accuracy versus the cost parameter. The weights were set to the figurative and literal class priors in the training data which in our case were 0.51 and 0.49 respectively. We obtained the highest accuracy of 0.779831932 using $C = 0.59$. When we combined all the features

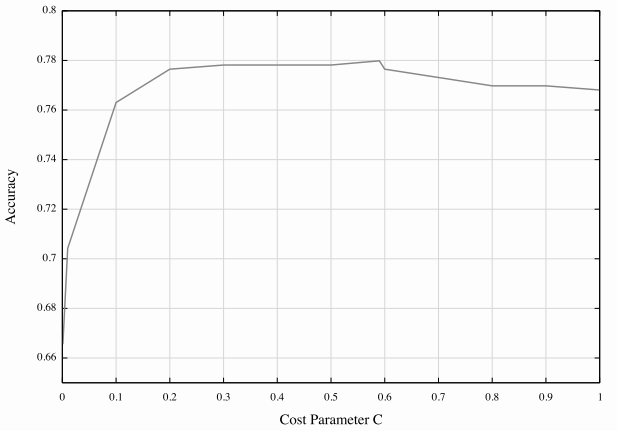

Figure 1: Plot of Accuracy versus parameter C

discussed in Section 2, we obtained an overall accuracy of 0.7680 as opposed to using only some features at a time, we obtained a better accuracy. This is because many of our features were overfitting the training data. The final accuracies reported only consider bag of words and semantic features - Phrase to Context Relatedness and Context Words Relatedness which gave us a total of $12,647$ features. We analyze this phenomena in Section 4.2.

Table 2 compares our results with those obtained by teams participating in the competition. The baseline is obtained by labeling all the test instances to the majority class in training. It can be observed that our accuracy is only marginally better than the second best. We obtained a $F1 = 0.776$ score for the literal class and $F1 = 0.783$ score for the figurative class.

Table 2: Comparison of our results with other teams. NAED represents our accuracy.

Participant	Accuracy
NAED	0.779831932
IIRG 3	0.779461279
UNAL 2	0.754208754
UNAL 1	0.722222222
IIRG 1	0.53030303
BASELINE	0.50337
IIRG 2	0.501683502

Figure 2 displays the "macro-average" accuracy and $F1$ scores for the classes for each phrase.

4.2 Discussion

The fact that our overall accuracy is only marginally better than the second best indicates that this is almost the best accuracy that can be obtained by training and testing on the given corpus. The $F1$ scores reported indicate that our algorithm can predict figurative and literal with almost equal accuracy but it is slightly better on predicting figurative. This may be due to slightly more number of figurative instances in training and also that our features were directed towards identifying figurative and classifying the others as literal.

While experimenting and analyzing various features extracted from data, it was observed that bag of words by itself proved to be significantly accurate. This is because the kind of words and their frequency are different for literal and figurative instances. The $tf - idf$ feature and skip(gappy) bigrams features helped in marginally boosting the bag of words features but did not affect he accuracy much. The semantic similarity features proved to be very useful because they indicate that unigrams in literal instances are more semantically similar to phrases than unigrams in figurative instances. As opposed to semantic similarity, the co-occurrence feature did not help in improving the accuracy. The reason is that the terms that co-occurred in training did not necessarily co-occur in test. The polarity feature marginally improved the accuracy but when combined with semantic features was hurting the accuracy and thus we did not include it in our final results. The concreteness feature unexpectedly did the not improve the accuracy, the reason being the list of words provided by the MRC Database was very limited and only few of those words actually occurred in our corpus.

For the macro-average case, we obtained maximum literal $F1$ score for the phrase "under the microscope" and the maximum figurative $F1$ score for the phrase "rub it in". It is not hard to tell that the former phrase almost always occurs in a literal sense while the latter one is almost always used in a figurative sense and thus we were able to extract better features and learn well on these phrases. For a phrase like "play ball" wherein the figurative and literal $F1$ scores are almost equal to the accuracy, it is hard to distinguish the sense of phrases because the two senses of the words are homonyms of each other.

5 Related work

Most of the work that has been done in idiom classification is based on the type-extraction method. As mentioned in (Li and Sporleder, 2010), this method focuses on the fact that idioms have properties that help differentiate them from other expressions such as a degree of syntactic and lexical fixedness. One example is that they are not used in a passive voice, for example *the bucket was kicked* instead of *kick the bucket*.

In addition, although there has been research done in Linguistics about the properties of idioms, there is still no agreement on which properties characterize them. Moreover there has been work done in trying to identify those properties automatically (Fazly et al., 2009). Nevertheless, in this approach they focused on a particular type of idioms which are formed by combining a frequent verb with a noun. Examples include *shoot the breeze* and *make a face*.

Numerous studies have focused on the Token-Based approach. One example is a minimally supervised algorithm that distinguishes if the sense in which a verb is being used is figurative or literal (Birke and Sarkar, 2006). Since it is focusing on the verb, which is a particular token, this approach does not consider the linguistic properties previously mentioned.

6 Future Work

We would like to further explore the impact of different combinations of features and to study the effect of one feature on another when use din unison. In addition, for our word relatedness measurements, we used the cosine similarity score, we would like to experiment more with other similarity measures. There is scope for future work in measuring word relatedness by comparing the degree of overlapping among words. This might have an impact in three of our current features that make use of word relatedness.

As for the topic extraction phase we had for one of the features, the Maui algorithm that we used performs better when the text is longer. A future enhancement would be to extend the current dataset with more literal and figurative context examples for each phrase and then extract new lists

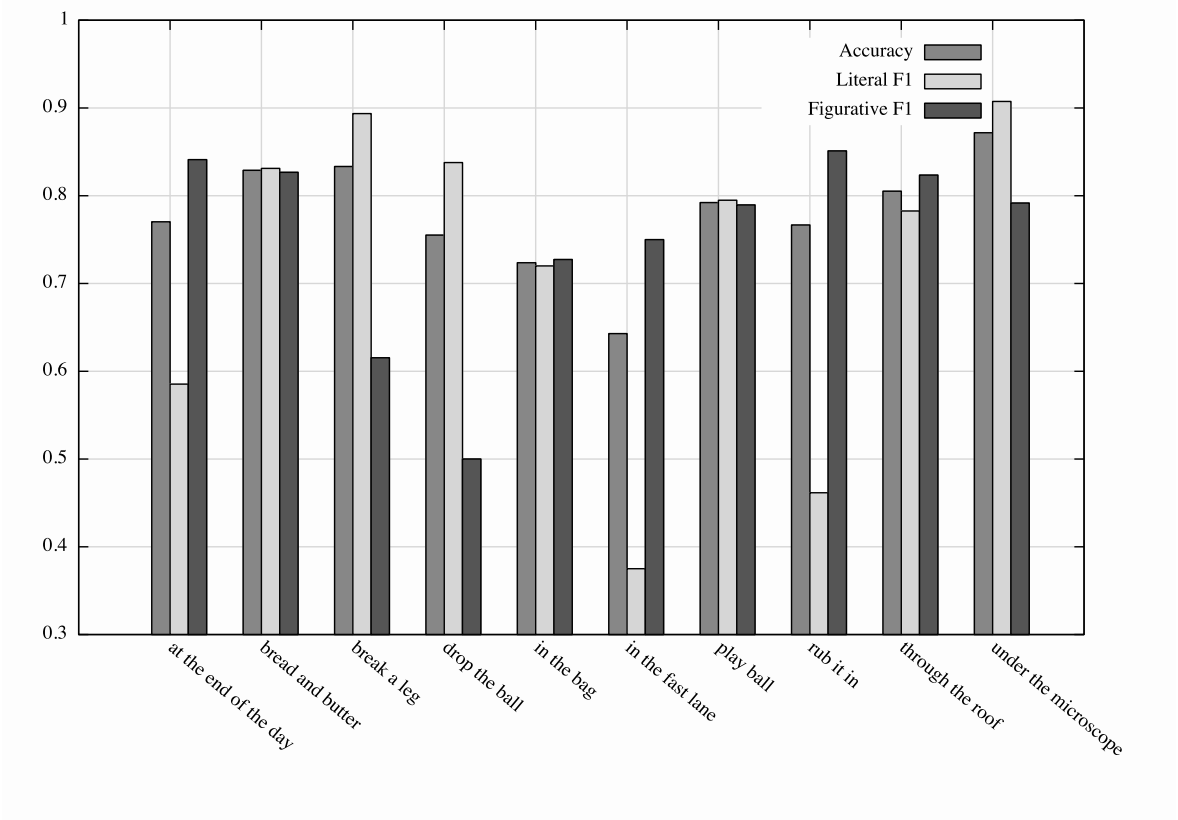

Figure 2: Macro-Average Accuracy and $F1$ scores

of topics. Better lists of topics would help us determine the sense of a phrase more accurately.

7 Conclusion

Metaphor is ubiquitous and recognizing figurative sense of phrases is a challenge in natural languages. An algorithm for distinguishing literal and figurative senses of a word will facilitate correct textual inference, which will improve many NLP applications that depend on textual inference such as machine translation, search, automatic summarization and even parts of speech tagging.

We introduced our algorithm that uses state of art features to predict if a given phrase has been used literally or figuratively. A strength of our algorithm is that since it only uses context in which the phrase appears, it can be easily generalized to phrases out of the training data. Our results outperform the existing systems on the same dataset and we believe that our approach has lot of scope in the aforementioned applications.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the 7th conference on International Language Resources and Evaluation (LREC10), Valletta, Malta, May*.

Julia Birke and Anoop Sarkar. 2006. A clustering approach for the nearly unsupervised recognition of nonliteral language. In *Proceedings of EACL*, volume 6, pages 329–336.

Max Coltheart. 1981. The mrc psycholinguistic database. *The Quarterly Journal of Experimental Psychology*, 33(4):497–505.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Afsaneh Fazly, Paul Cook, and Suzanne Stevenson. 2009. Unsupervised type and token identification of idiomatic expressions. *Computational Linguistics*, 35(1):61–103.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*, pages 47–54.

J. Goodman. 2000. A Bit of Progress in Language Modeling. Technical report, Microsoft Research, 56 Fuchun Peng.

Cyril Goutte and Eric Gaussier. 2005. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *Advances in Information Retrieval*, pages 345–359. Springer.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

Ioannis Korkontzelos, Torsten Zesch, Fabio Zanzoto, and Chris Biemann. 2013. Semeval-2013 task 5: Evaluating phrasal semantics. April.

Linlin Li and Caroline Sporleder. 2010. Using gaussian mixture models to detect figurative language in context. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 297–300. Association for Computational Linguistics.

Olena Medelyan. 2009. *Human-competitive automatic topic indexing*. Ph.D. thesis, The University of Waikato.

Ted Pedersen and Varada Kolhatkar. 2009. Wordnet:: Senserelate:: Allwords: a broad coverage word sense tagger that maximizes semantic relatedness. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Demonstration Session*, pages 17–20. Association for Computational Linguistics.

Caroline Sporleder and Linlin Li. 2009. Unsupervised recognition of literal and non-literal use of idiomatic expressions. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 754–762. Association for Computational Linguistics.

Peter D Turney, Patrick Pantel, et al. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188.

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 2011 Conference on the Empirical Methods in Natural Language Processing*, pages 680–690.

Dominic Widdows and Beate Dorow. 2005. Automatic extraction of idioms using graph analysis and asymmetric lexicosyntactic patterns. In *Proceedings of the ACL-SIGLEX Workshop on Deep Lexical Acquisition*, pages 48–56. Association for Computational Linguistics.