A Distributional Semantics Model for Metaphor Detection

Jasim Ahmed , Mirza Mustansar Baig

Department of Computer Science, Universität Passau

October 14, 2018

1. ABSTRACT

Metaphors are an integral part of any language. They bring life to the effective communication. Many approaches have been proposed to detect the sense of a phrase being used as literal or metaphorical. Metaphors are generally compositional and this composition can be translated to a source domain of a concept to a target domain of a different but related concept. Compositional distributed semantic models provide a good way to understand this composition and help in identifying the sense of a phrase to be used as literal or metaphorical. It is also important to know the context of a phrase which can help in figuring out the sense and this context can be analyzed using vectors. Using Distributional Semantic Models, phrases can be identified as being used metaphoric by applying algebraic expressions on vectors.

2. INTRODUCTION

Metaphors are an interesting way of communication. It is generally a comparison referring to a person, place or thing as being something else i.e "Your brain is a computer". In this example, brain is referred to as computer due to its capability to solve complex problems in less time. One of the literal ways of conveying this idea would be "You are a smart person with extreme mental capabilities". Metaphors can be divided into two categories, conceptual metaphors and linguistic metaphors. Linguistic metaphors are instantiations on the broader category of conceptual metaphors. While instantiating the linguistic metaphor, there is always a mapping that is performed from the source of the concept to the target of that concept. In the metaphor, "Your brain is a computer", the source domain of "Human Intellect" is mapped on the target domain of "Digital Computation". According to the Conceptual Metaphor Theory, this mapping is always done systematically and hence can be calculated in the sentences to figure out the sense being literal or metaphorical. Metaphorical sense is not the characteristic of an individual word, instead it is composed by the mapping of one domain to another domain.

The metaphorical senses of sentences have been detected by many techniques including Distributional Statistics(DS), Vector Space Models(VSM), Distributional Semantic Models(DSM), Compositional Distributional Semantic Models(CDSM) etc. An interesting characteristic of metaphors is that their metaphorical or literal sense can be detected by the way they are composed using source and target domain e.g "heavy processing", "heavy logic" are composed by interplay of source and target domains.

This compositionality can be determined by the compositional distributional semantic models (CDSM) in which phrases are treated as vectors. From these phrases, nouns are represented as vectors and adjectives as matrices. Due to the systematic composition of the phrases, the context of the phrases can also be computed using CDSM.

We are using the idea of adjective-noun phrase vectors. Once the vectors are created, these vectors are fed to the classifier as attributes. Based on these attributes, classifier, classifies the phrase and annotates the phrase as being used in literal or metaphorical sense.

Figure 1: An example of Metaphor

3. Similar Ideas

Two similar ideas exist along with the metaphors in phrase compositions. These are as follows:

3.1 Similie

It refers to a comparison that is done using the word "as" e.g "as bright as a star".

3.2 Idioms

It refers to an expression that means something else than the words say.

3.3 Morphism

The structure of the metaphors can also be understood by the idea of "morphism" which is a concept given by "Category Theory". Morphism is the transformation of an object to another object in such a way that it preserves some of the structure of original object. This transformation of objects can be explored to understand the composition of the metaphor.

3.4 Polysemy

A polyseme is a word or phrase with different, but related senses. This property of the word introduces ambiguity while understanding it and can be done if the context of the word is given. CDSMs can avoid this ambiguity while making sense of a word since it contains matrices for adjectives which already contain the context of the word.

4. Background

4.1 Distribution Theory

The distributional hypothesis in linguistics is that words that occur in similar contexts tend to have similar meanings (Harris, 1954). This hypothesis is the justification for applying the DSM to measuring word similarity. A word may be represented by a vector in which the elements are derived from the occurrences of the word in various contexts. If words have similar row vectors in a word context matrix, then they tend to have similar meanings.

4.2 Compositional Distributed Semantic Model

The main idea behind the CDSMs is to perform algebraic expressions on the vector representation of words and phrases and from those expressions, learn the composition of phrases. With this knowledge about the composition, we can detect whether the phrase is used in literal or metaphorical sense.

4.3 Krishnakumaran and Zhu Metaphor Detection

They focused on detecting whether an adjective-noun phrase is used in a literal or metaphorical way. They kept the count of adjective-noun co-occurrence in the dataset and the WordNet hyponym and hypernym relations. The main idea is that if some certain noun and hypernym/hyponym do not occur very frequently with some certain adjective then this phrase is used as a metaphor.

5. Main Approach

Our main approach is inspired by all the ideas above. Following is the main activity/flow diagram which describes the overall structure of the approach taken.

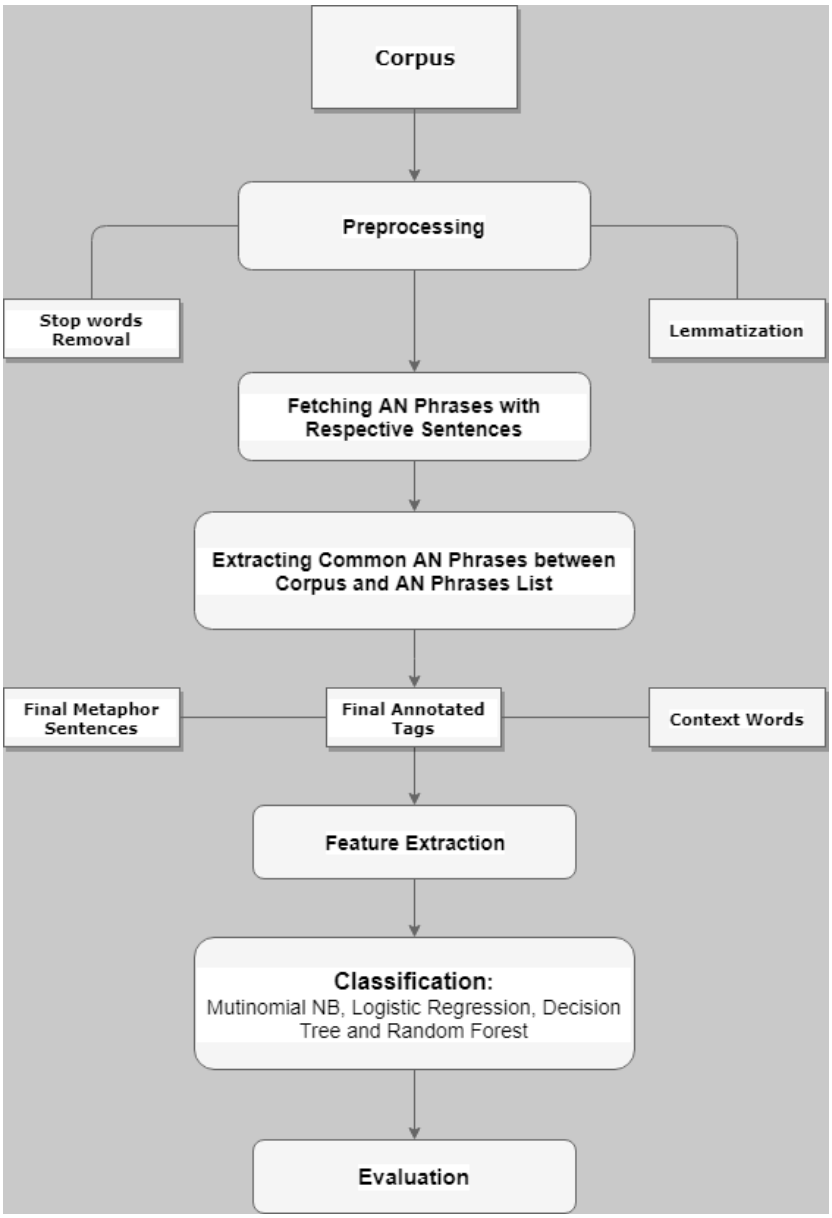

Figure 2: Flow Diagram

5.1 Corpus

We chose Brown corpus from NLTK. It contains different kinds of categories including Adventure, Editorial, Fiction, Government, Hobbies, Humor, Learned, Lore, Mystery, News, Religion, Reviews, Romance and Science Fiction. Many of these categories contain phrases that are used in metaphorical senses.

5.2 Pre-processing

Brown corpus is already POS tagged so we did not have to POS tag it on our own. POS tagging helps in finding the Noun and Adjectives which will be used in order to annotate the phrases later on. For simplicity, we are working with the universal tag set which simplifies and limits the tags to their very basic form.
In order to save processing time, stop words are removed from the corpus sentences at the first step. After that these sentences are lemmatized to save the processing time even further. In the next step, all the adjective noun(AN) phrases are extracted from the sentences. This is created into a tuple having the sentence as the first element and the AN-phrase as the second element of the tuple. the reason we are extracting the AN-phrase from the sentences is that we will use these phrases later on to extract all the matching phrase sentence from the corpus.
Now that we have all the phrases and their respective sentences, we can compare these phrases with the AN-phrase set and only take the matching phrase sentences. The rest of the sentences are discarded. This step significantly reduces the our matrix computation later on because the number of sentences are reduced to only the desired AN-phrase matching set. While calculating the common sentences between the corpus sentences and the AN-phrase set, sentences are annotated and context words are fetched at the same time. Since the Brown corpus is very large and requires a significant amount of processing time in order to look at all the phrases and decide whether it is used as literal or metaphor, we fetched all the common words and phrases from the corpus. These phrases are then fed to term-document matrix afterwards for generating the vectors for the phrases.

5.3 AN-Phrase Dataset

We are using a data set of adjective-noun phrases. Each phrase in the data set was annotated manually by a group of people as being used either literally or metaphorically. This data set was created by E. Dario Gutierrez, Ekaterina Shutova, Tyler Marghetis and Benjamin K. Bergen. The data set was created by choosing 23 adjectives that can possibly have both literal as well as metaphorical senses. Each of these adjectives behave as source-domain words while constructing conceptual metaphors described by Conceptual Metaphor Theory. The annotated data set is publicly available at http: //bit.ly/1TQ5czN

5.4 Sentence Annotation(Metaphor/Literal)

The common extracted sentences need to be annotated as literal or metaphorical. This annotation is the key to classification. A sentence is annotated based on the annotation of the respective AN-Phrase. If the AN-Phrase has metaphorical sense annotation then the sentence is annotated as "Y" otherwise it gets annotated as "N".

5.5 Context Words

The context words are the words that are obtained by removing the AN-Phrase from the actual corpus sentence. The remaining words are the words that are

used as the context of that particular AN-Phrase. This approach will help us making the co-occurrence matrix with exact context of a AN-Phrase.

5.6 Feature Extraction

Based on the above mentioned AN-phrase data set, we extracted all the sentences from the data set that involve these phrases. With this step, we reduced the processing time of creating the vectors for the phrases from the Brown corpus. Without this, we had to check all of the sentences and construct the matrices afterwards. Now we can only work with these selected sentences and find out about their usage sense. After finding all the relevant phrases along with their annotation tags and context words from the corpus, a document-term matrix was constructed. In this matrix, the documents are the sentences that are extracted based on the AN phrase set and the terms are the common words in the whole corpus. The resulting matrix is the collection of frequency vectors. Each vector represents a sentence(document) which is either used in literal or metaphorical form. This vector inherently contains all of the context of the phrase that needs to be looked at for its sense understanding.

5.7 Classification

The classification technique is a systematic approach to build classification models from an input data set. For example, decision tree classifiers, rule-based classifiers, neural networks, support vector machines, and naive Bayes classifiers are different technique to solve a classification problem. Each technique adopts a learning algorithm to identify a model that best fits the relationship between the attribute set and class label of the input data. Therefore, a key objective of the learning algorithm is to build predictive model that accurately predicts the class labels of previously unknown records. Since the data we are working on is multi-label data, one of the obvious choices for the classifier is Multinomial Naive Bayes classifier.

5.7.1 Naive Bayes

The Naive Bayes Classifier technique is based on the so-called Bayesian theorem and is particularly suited when the dimensionality of the inputs is high. Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods.

To demonstrate the concept of Naïve Bayes Classification, consider the example displayed in the illustration above. As indicated, the objects can be classified as either GREEN or RED. Our task is to classify new cases as they arrive, i.e., decide to which class label they belong, based on the currently exiting objects.

Since there are twice as many GREEN objects as RED, it is reasonable to believe that a new case (which hasn't been observed yet) is twice as likely to have membership GREEN rather than RED. In the Bayesian analysis, this belief is known as the prior probability. Prior probabilities are based on previous experience, in this case the percentage of GREEN and RED objects, and often used to predict outcomes before they actually happen.

Since Naive Bayes uses clustering technique, this technique is useful for metaphor

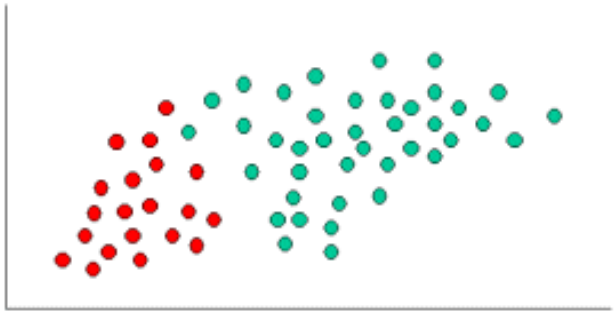

Figure 3: Naive Bayes

classification. As the vectors are given as the X component and annotation tags are given as the Y component, Naive Bayes clusters the literal vectors in in one cluster and metaphor classifier in another cluster, hence classifies the sentences.

5.7.2 Decision Tree

The decision tree classifiers organized a series of test questions and conditions in a tree structure. The following decision tree is for predicting whether the person cheats. In the decision tree, the root and internal nodes contain attribute test conditions to separate records that have different characteristics. All the terminal nodes are assigned a class label either Yes or No.

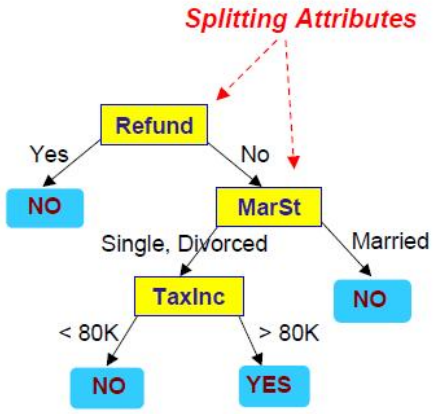

Figure 4: Decision Tree

Once the decision tree has been constructed, classifying a test record is straightforward. Starting from the root node, we apply the test condition to the record and follow the appropriate branch based on the outcome of the test. It then leads us either to another internal node, for which a new test condition is applied, or to a leaf node. When we reach the leaf node, the class label

associated with the leaf node is then assigned to the record. As shown in the following figure it traces the path in the decision tree to predict the class label of the test record, and the path terminates at a leaf node labeled NO.
We chose the decision tree classifier because its works on the binary decision of to be or not to be. We need the classifier to classify whether a certain phrase is used in literal or metaphorical sense.

5.8 Random Forest

Random Forests are an ensemble of k untrained Decision Trees (trees with only a root node) with M bootstrap samples (k and M do not have to be the same) trained using a variant of the random subspace method or feature bagging method. The procedure for training a random forest is as follows:

1. At the current node, randomly select p features from available features D. The number of features p is usually much smaller than the total number of features D.

2. Compute the best split point for tree k using the specified splitting metric (Gini Impurity, Information Gain, etc.) and split the current node into daughter nodes and reduce the number of features D from this node on.

3. Repeat steps 1 to 2 until either a maximum tree depth l has been reached or the splitting metric reaches some extrema.

4. Repeat steps 1 to 3 for each tree k in the forest. Vote or aggregate on the output of each tree in the forest.

Random Forest is a special case of Decision Tree and can be used for the metaphor detection as explained earlier.

6. Case Study

Let's take an example to better understand how the approach works with a case study. Let for example consider the following sentences as the Corpus which needs to be classified.

- S1 = "Simon is a straight-A student."
- S2 = "Simon is a bright boy."
- S3 = "Simon's shoes have bright colors."
- S4 = "Simon lives near Passau."

S1, S2, S3 and S4 represent documents in a corpus. Let us also consider the following AN phrase set:

- P1 = "bright boy"
- P2 = "bright color"

Where P1 and P2 represent the phrases from an already created AN-Phrase (Adjective Noun Phrase) set. The AN-Phrase set contains phrases along with their senses being either metaphorical or literal. In the first step of preprocessing, we take all the sentences and remove all the stop words from them. After the stop words removal and lemmatization the sentences look as follows:

- S1 = "Simon straight-A student."
- S2 = "Simon bright boy."
- S3 = "Simon shoe bright color."
- S4 = "Simon live near Passau."

In the next step, only the sentences that involve P1 and P2 are kept for further processing and the remaining data will be discarded. In our case study, since only S2 and S3 match with P1 and P2, so we keep these and discard S1 and S4.
Now we need to annotate S2 and S3 and this annotation will be used for the classification step. These sentences are annotated according to the annotation of P1 and P2 which was given in the AN-Phrase data set. After the preprocessing of the raw data, we are all set to create a word-context(document-term) co-occurrence matrix. This matrix is created using tf-idf technique which will put the S2 and S3 on the left hand side and all the context words within these sentences on top of the matrix. The matrix looks like the following:

	Simon	Shoe	Bright	Color	Boy
Simon bright boy	1	0	1	0	1
Simon shoe bright color	1	1	1	1	0

Figure 5: TF-IDF Matrix

This matrix is then fed into the classifier which then learns the vectors based on the annotation tags and classifies the test vectors as being literal or metaphorical. In our case study, the classifier learns that the metaphorical vectors will look like the following:

1	0	1	0	1

Figure 6: TF-IDF Metaphorical Vector

And the literal vectors will look as below:

1	1	1	1	0

Figure 7: TF-IDF Literal Vector

Of course in real time classification, the data is always huge and classifier needs time before it learns all the patterns of both literal and metaphorical senses from the train data.

7. Evaluation and Results

In this section we have shown all the results after applying different kinds of classifiers. These results have been displayed on Bar Charts showing the comparative accuracy, precision, recall and F1 scores.

7.1 Train Test Split

The Data in "Word Context Matrix" is split into training data and test data using "Train Test Split". The mean accuracy for Multinomial Naive Bayes Classifier and Decision Tree is shown in the following Figure.8

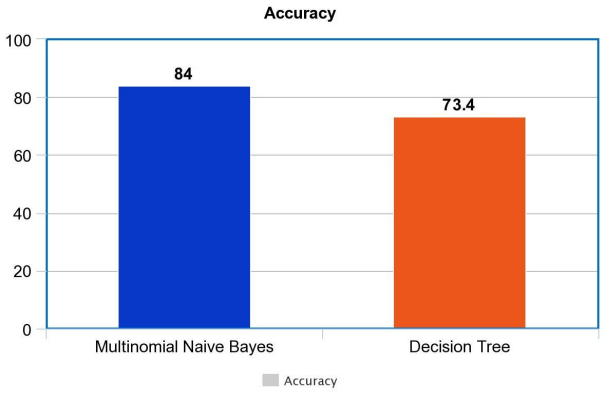

Figure 8: Classifiers

7.2 Repeated K-Fold

The Data in "Word Context Matrix" gets split into k subsets, and trained on one of the k-1 subset and the last subset is held for test. Repeated K-Fold is used in estimating whether our model is over-fitting or under-fitting.
Applying repeated K-Fold on four different classifiers, the mean accuracy varies a lot on given test data and labels. As shown in Figure 9

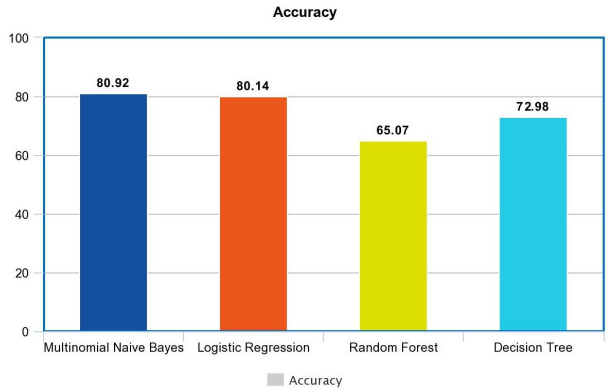

Figure 9: Accuracy

The comparative mean precision is shown in the following Figure 10

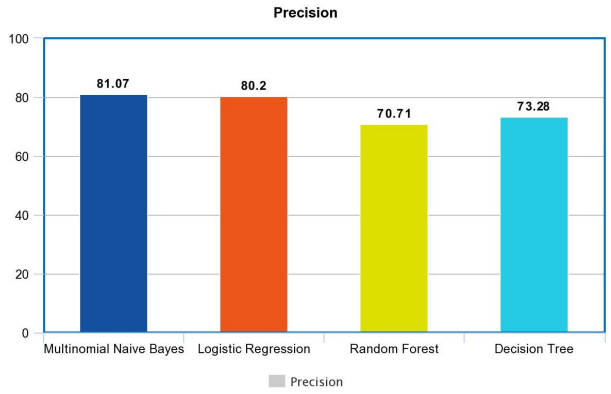

Figure 10: Precision

The mean Recall is shown in the following Figure 11

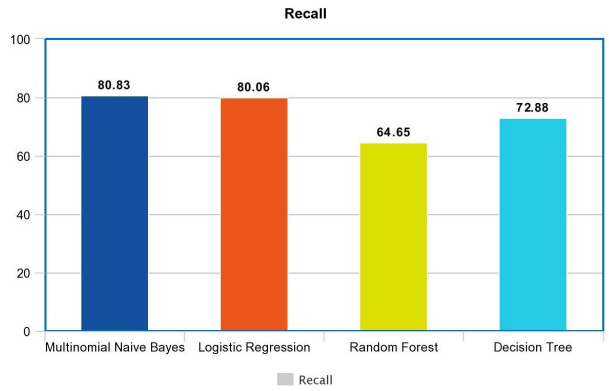

Figure 11: Recall

The mean F1_score is shown as in Figure 12

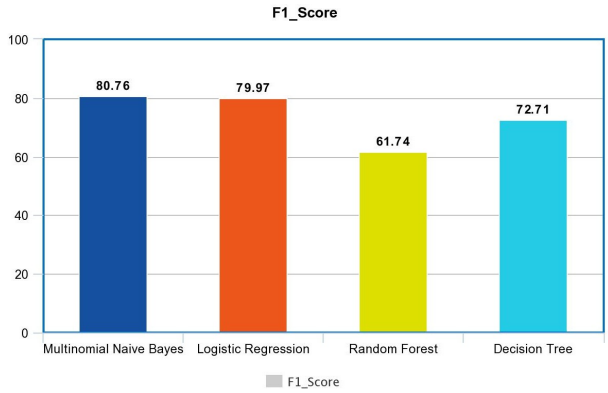

Figure 12: F1_Score

8. Conclusion

Distributed Semantic Models provide an approach to find the metaphors in a document(Sentence) based on the context of a phrase that needs to be classified as literal or metaphorical. This paper features an approach to classify a sentence from a set of corpora to be either literal or metaphorical. Features are extracted based on TF-IDF extraction technique. In order to perform the classification, four different classifiers are used namely "Multinomial Naive Bayes", "Decision Tree", "Random Forest" and Logistic Regression(Multinomial) were used. Our experimental results show that out of these classifiers, Multinomial Naive Bayes stands out with accuracy of 84%.

References

[1] Literal and Metaphorical Senses in Compositional Distributional Semantic Models
E. Dario Gutierrez, Ekaterina Shutova, Tyler Marghetis, Benjamin K. Bergen, University of California San Diego, University of Cambridge, Indiana University Bloomington]

[2] From Frequency to Meaning: Vector Space Models of Semantics
Peter D. Turney peter. National Research Council Canada Ottawa, Ontario, Canada, K1A 0R6

[3] Feature Extraction TF-IDF TF-IDF stands for “Term Frequenct—Inverse Data Frequency”

[4] Naive-Bayes-Classifier The Naive Bayes Classifier technique is based on the so-called Bayesian theorem and is particularly suited when the dimensionality of the inputs is high.

[5] Train Test Split and Cross Validation In K-Folds Cross Validation we split our data into k different subsets (or folds)

[6] Basic-Evaluation-Measures The confusion matrix is a two by two table that contains four outcomes produced by a binary classifier

[7] Decision Trees The classification technique is a systematic approach to build classification models from an input dat set

[8] Saisuresh Krishnakumaran and Xiaojin Zhu. 2007. Hunting elusive metaphors using lexical resources. In Proceedings of the Workshop on Computational approaches to Figurative Language, pages 13–20. Association for Computational Linguistics.]

[9] Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 1183–1193. Association for Computational Linguistics.

[10] M. Baroni, S. Bernardini, A. Ferraresi, and E. Zanchetta. 2009. The WaCky wide web: A collection of very large linguistically processed web-crawled corpora. Language Resources and Evaluation

[11] Marco Baroni, Raffaela Bernardi, and Roberto Zamparelli. 2014 Frege in space: A program of compositional distributional semantics. Linguistic Issues in Language Technology.

[12] Marco Baroni, Raffaela Bernardi, and Roberto Zamparelli. 2014 Random Forests are an ensemble of k untrained Decision Trees (trees with only a root node) with M bootstrap samples.