Extracting Character Relationships From Stories

Krishna Janakiraman

Abstract

In this report, I have detailed my experiments in developing an unsupervised learning algorithm for extracting the relationship between a pair of characters in a story. My algorithm uses the 'bag of words' model and the correlation coefficient to group related character pairs across a corpus of stories. The relationship for a group of character pairs is then determined by computing the semantic similarity score between the words associated with the pairs and terms from a relationship vocabulary. I have evaluated my algorithm on a corpus of short stories against three different vocabularies. I was able to achieve a maximum precision rate of 55% for a reasonably sized vocabulary.

1 Introduction

Machine understanding and appreciation of story has been among the most cherished goals in Natural Language Processing research. A key step towards understanding a story is to understand the relations between the characters that occur in the story. This problem can be thought of as a special case of the traditional relation extraction problem where, the task is to identify the relationship between any two general entities occurring in a text. Relation extraction problems are solved either through supervised learning or unsupervised learning algorithms.

In supervised learning, we have access to a corpus of text for which the entities and their relation types are already known. Such algorithms typically learn to classify new entity pairs into any of the relation types it has already seen based on some re-occurring patterns.

Unsupervised learning approach is used when we do not have access to such a marked up corpus. Such algorithms typically identify patterns, relevant to the relation extraction task, occurring within the corpus and then use these patterns to group entities such that entities within a group share similar relationships. Unlike the supervised approach where the relation types are already known, a major challenge in the unsupervised approach is to generate a relationship term for a group of entities.

For my final project, I have designed and experimented with an unsupervised learning algorithm that extracts a relationship term for a given pair of characters in a story. My algorithm uses a 'bag of words' approach to associate a set of words to a character pair; then, the algorithm groups similar character pairs based on the correlation coefficient between the words associated with them. Relationship term for a group of pairs is then determined by computing the semantic similarity score between the 'bag of words' associated with the group and a set of relationship terms taken from a relationship vocabulary.

2 Related Work

In a recent work, Jurafsky et al (2009) proposed a supervised approach for extracting relationship between generic entities using Freebase as the corpus. Freebase lists thousands of relations between million different entities. Instances for these relations were taken from respective Wikipedia entries that mention the entities. Their proposed method uses both lexical and syntactic features computed over a

region surrounding the sentences the entities occur.

Marti Hearst and Barbara Rosario (2004) propose a supervised algorithm that can identify the relation between a disease and a treatment using bioscience texts as the corpus. Their method uses a combination of lexical, syntactic and semantic features. They defined a relationship vocabulary that listed seven possible relations between disease and treatment and used both generative and discriminative models to build their classifiers.

Goto et al (2009) propose an unsupervised method to generate graphs showing relationship networks among characters occurring in program summaries. Their method uses a combination of standard name entity recognition methods and rules based anaphora resolution to identify the characters in the summary. Their relationship terms are extracted directly from the text using syntax analysis.

3 Data

3.1 Corpus

The corpus consists of a set of 55 short stories taken from Project Gutenberg. To cover a fair variety of linguistic styles, the stories were taken from a number of authors including R. L Stevenson, O Henry, Sir Arthur Conan Doyle and could be categorized broadly into the following genres : Fiction, Children Stories, Summaries of Epics.

3.2 Extracting Characters

A fundamental step towards identifying relationship between character pairs is to identify the characters themselves. This in itself is a non trivial problem and was beyond the scope of my experiments. Therefore, this task was accomplished using the Name Entity Recognizer toolkit from Stanford (Finkel, Grenager and Manning 2005).

The Name Entity Recognizer toolkit parses a text and marks up the following entities occurring in the text : *Person*, *Organization* and *Location*. All the words marked up as *Person* by the toolkit were taken as characters in the story. The Name Entity Recognizer was already trained on a large corpus and was fairly accurate in identifying characters that were proper nouns. However, in some stories like Cinderella for example, the prince and the fairy are legitimate important characters but are not proper nouns and hence could not be marked up by the entity recognizer. Thus, for a small subset of stories, the output of the toolkit was manually edited so that important characters are marked up.

3.3 Relationship Vocabulary

The actual relationship term was picked from a relationship vocabulary based on a semantic similarity score. The following three different vocabularies were used and compared against each other:

Vocabulary1: *friend, rival*

Vocabulary2: *parent, child, friend, sibling, ancestor, rival, spouse, mentor, nephew*

Vocabulary3: *son, daughter, friend, colleague, brother, sister, rival, husband, wife, teacher, student, mother, father, grandparent, aunt, uncle, cousin*

The vocabularies can be seen as increasing in their ability to discriminate. The motive was to experiment whether a smaller, much less accurate, vocabulary performed better.

4 Features

My features are similar to the lexical features used by Jurafsky et al and are based on the hypothesis that words that surround a character pair describe the relationship between the pair.

The features are computed using 'bag of words' collected from a window defined around the sentences containing the character pair. Words nearer to the pair were given higher weight and words away from the pair were penalized according to their distance from the pairs. The following sequence of steps explain the feature computation process for a given character pair:

1. Find all sentences in the story that the pair occurs in.

2. For each of the above occurring sentence, collect k sentences to its right and left.

3. Tokenize each sentence into words. Normalize the words and lemmatize the words using WordNet. Note that this essentially discards

words that are not present in the WordNet lexicon. I also discarded stop words using the stop word list provided by the Python NLTK tool kit.

4. Maintain a distance score $ds_{pair,word}$ for each pair, word combination. If a word occurs in the same sentence, give it 1 point, if it occurs in the k^{th}sentence from the occurring sentence, give it $\frac{1}{2^k}$ points. These points are accumulated every time a word occurs in the defined window.

From the 'bag of words' collected for each pair, we form a Character Pair - Words matrix, M. For the training corpus of short stories, the number of pairs was 495, and the number of unique words varied between 2750-2850 depending on the sentence window size.

The Character Pair - Words matrix is similar to the Document-Term matrix that is typically computed for Information Retrieval tasks and the term frequency $tf_{word,pair}$ and the inverse document frequency idf_{word} scores were similarly computed:

$$tf_{pair,word} = \frac{no\,of\,times\,word\,is\,seen\,with\,the\,pair}{total\,no\,of\,words\,seen\,with\,the\,pair}$$

$$idf_{word} = \log \frac{total\,no\,of\,words}{no\,of\,times\,word\,has\,occured}$$

Each entry in the matrix was then computed using the following formula :

$$M[pair][word] = ds_{pair,word} * tf_{pair,word} * idf_{word}$$

Intuitively, this formula denotes that important terms that are closer to the pair are weighted higher.

5 Training

The goal in the training phase is to group character pairs based on a similarity measure computed on the features. The motive is that, while prediction, such a grouping would provide us with a richer set of words to compare with the terms in the relationship vocabulary.

To group the characters, a neighborhood model was built for each of the character pair using the Pearson's correlation coefficient . Given the word vectors for two character pairs X, Y and μ the mean word vector computed across all the pairs, their Pearson correlation measure is given by the following formula :

$$COR(X,Y) = \frac{E[(X-\mu)^t(Y-\mu)]}{\sqrt{Var(X)Var(Y)}}$$

This correlation coefficient measure ranges from -1 to +1. Since this measure is computed directly from our features, character pairs with a high correlation score are highly likely to share the same relationship and therefore the 'bag of words' associated with one pair in the group can be used towards predicting the relationship term for the other pair.

6 Prediction

For a given character pair and a relationship vocabulary, the relationship terms from the vocabulary are ranked as follows :

1. Form a neighborhood of n pairs closest to the test pair using the correlation measure

2. For each pair in the neighborhood, find the words associated with it :

3. For each word, find its score from the Character Pair - Words matrix :

   (a) For each relationship term in the vocabulary, compute WordNet path_similarity measure between the word and the relationship term

Using the above procedure, we accumulate the similarity measure between each word in the 'bag of words' formed by the neighborhood and a relationship term. We then rank the relationship terms based on their accumulated similarity measure.

The below table shows the top five relationship terms that was predicted for the character pair Rama and Sita from the Indian epic Ramayana. The relation was correctly predicted as *Wife/Husband*.

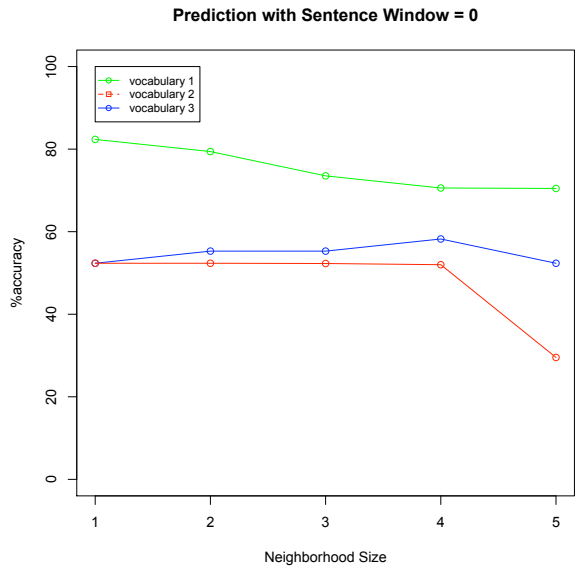
Relation	Score
Wife	18.2
Friend	7.2
Husband	6.7
Student	6.2
Son	4.3
Mother	2.4

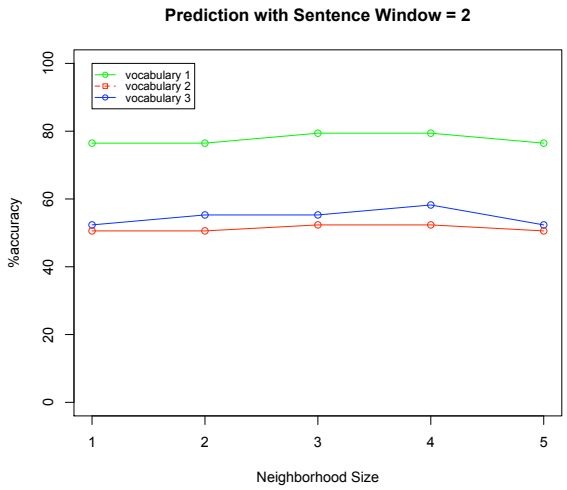
7 Testing

For testing, a list of 100 character pairs were created. The relation between the pairs was manually determined. The pairs were selected from popular stories so that their relation is well known and hence reading effort could be minimized. To circumvent the problem of non symmetric relations like *mother* and *daughter*, both the relations where mentioned. A success was assumed if one relation was predicted accurately.

7.1 Results

The results are summarized by the graphs shown below. Each graph shows the accuracy percentage for the three vocabularies using training done on different sentence window sizes.





1. We see that using a sentence window does not play a major part in the final results,contrary to what was expected initially. This may also have to do with the weighting scheme that was used to compute the distance score between a word and a pair.

2. Having a neighborhood model improved the results. In both the graphs I got the best results for a neighborhood of size 4.

3. As expected, the algorithm performs well on vocabulary1 as it defines only two relations. But such a vocabulary will be too ambiguous to be useful.

4. Though vocabulary2 is smaller than vocabulary3 it performed poorer. The reason could be that terms in vocabulary 3 are much more specific than those is vocabulary 2, for example : *wife, husband* compared to *spouse*, and thus have a higher probability of a direct match with the words occurring in the Character Pair - Words matrix, thus giving them a high score.

8 Conclusions

In this report, I have described an unsupervised algorithm for extracting relationships terms between a pair of characters from a story. I evaluated my algorithm on a corpus of short stories using three different vocabularies. My results suggest that a simple 'bag of words' based features is not adequate enough for this task. However, the results show that the bag of words model perform well towards the

task of grouping similar character pairs in a corpus of stories. Combining the correlation based neighborhood model with syntactic features should yield better results.

References

Mike Mintz, Steven Bills, Rion Snow, Dan Jurafsky. *Distant supervision for relation extraction without labeled data*. Proceedings of ACL-IJCNLP 2009

Barbara Rosario Marti A. Hearst. *Classifying Semantic Relations in Bioscience Texts*. ACL 2004

Juan Goto. *Method for Automatically Generating Networks of Personal Relationships from Story Summaries*. Common Sense and Intelligent User Interfaces 2009

Project Gutenberg. http://www.gutenberg.org/wiki/Main_Page

Python NLTK Toolkit. http://www.nltk.org/

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. *Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling*. Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370

WordNet: An Electronic Lexical Database. ISBN-10: 0-262-06197-X