

Semantic Taxonomy Induction From Semi-structured Text

Jui-yi Kao, Yi-hao Kao, and Ian Yik Oon Quek

Advised by: Daniel Jurafsky and Rion Snow

1 Introduction

Many computational linguistics and natural language processing tasks require a large lexical database of word relations. An interesting direction to explore in this line of research is to use more structured texts such as dictionary definitions and encyclopedia entries. These data sources have much more consistent structures that can be exploited by a computer to recognize word relations. Dictionary definitions, in particular, may also contribute to the recognition of certain groups of relation instances that are difficult to come by in general texts. They can also have a much more uniform coverage because each strives to be comprehensive.

In this project, we investigate applying automatic taxonomy induction techniques as presented in [1,2] to more structured texts. In particular, we show using Merriam-Webster dictionary definitions that applying a priori knowledge of the text structure vastly improves results over treating the corpus as free text. Our general approach is a variation of the approach taken by Snow, Jurafsky and Ng [2], adapted to the specific semi-structured text corpus we consider.

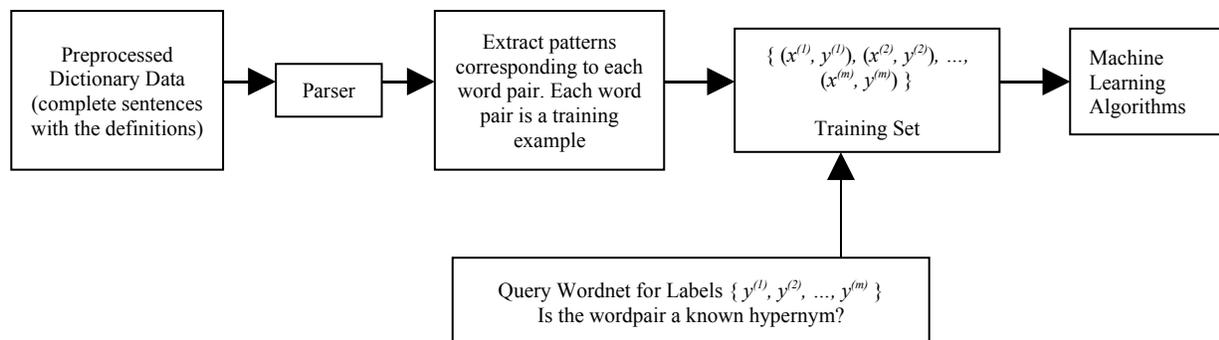


Figure 1: High level flow chart of our process.

2 Treating dictionary as free text

2.1 Preprocessing dictionary data

Before utilizing the data from the dictionary, we have to rewrite it into a useful sentence structure, so that the Stanford parser will be able to parse the sentence effectively. Dictionary data is typically in incomplete sentences, with the word itself left out of the sentence. We want to include this word in the sentence, so that we can use the parser to detect word pair relations from the sentence structure. We have observed some rules to preprocess the dictionary data:

- If the noun ends in “-phy,” “-ing” or “-tion,” we append “<word> is [the]” to the beginning of the definition. For example, “Absorption is the process of absorbing or of being absorbed.”
- If the definition starts with “the,” we append “The <word> is” to its beginning; if the definition starts with “an” or “a,” we append “[A/An] <word> is.”

- If the meaning of a noun does not have “the”, “an” or “a” in front of it, we add “the” to the beginning of the definition before appending the word.

2.2 Generating attribute vectors: discovering patterns

The following procedure is used to generate an attribute vector for each word pair under consideration.

- Parse the text using Stanford Parser.
- Order of words in a list should be ignored because the order has no semantic significance. We distribute all relationships across conjunct relations [1]. For example, in the sentence “I love fruits like apples and peaches,” one would like the extracted pattern relating fruits to apples to be the same as the extracted pattern relating fruits to peaches. So distributing across conjunct relations, the typed dependencies:

```
prep_like(fruits-3, apples-5)
conj_and(apples-5, peaches-7)
```

are replaced with the typed dependencies:

```
prep_like(fruits-3, apples-5)
prep_like(fruits-3, peaches-7)
```

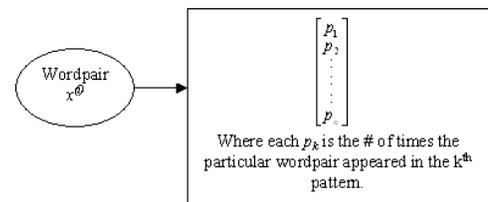


Figure 2: An attribute vector.

- Construct a dependency tree from the collapsed dependencies.
- Identify all distinct noun pairs related by a short enough dependency path. For each of these pairs of words, identify the minimal paths in the parse tree that contains instance of both words.
- Record the number of times the pair is related by each pattern throughout the whole text corpus. This is the set of attributes of each word pair considered.

3 Applying a priori knowledge of the text structure

To apply a priori knowledge of the text structure we do not form each defined word and its associated definition into a sentence, as presented in subsection 2.1. The same procedure outlined in subsection 2.2 is followed, with the following exceptions:

- In parsing, only the definition is parsed. The word being defined is left out. E.g. In the definition “weekend : the end of the week,” only the phrase “the end of the week” is parsed.
- After a dependency tree is constructed from the collapsed dependencies of the parsed definition, the word being defined is attached as the new root by introducing a special METADEFINE dependency and making each original root child to the word being defined through this METADEFINE dependency.

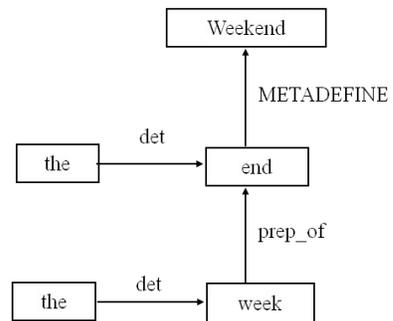


Figure 3: Adding the METADEFINE dependency.

4 Experimental paradigm

The goal of this project is to build a classifier that decides, given an ordered pair of nouns (n_i, n_j) , whether it is related by hyponym-hypernym relationship.

The experiments are based on a corpus of approximately 40 thousand dictionary definitions in the Merriam-Webster dictionary.¹

60,449 of the unique word pairs extracted by the definitions were labeled as Known_Hypernym or Known_NonHypernym using WordNet.² For each noun pair (n_i, n_j) in our data set, we label it as $Y((n_i, n_j))$, which is a binary value defined by:

$Y((n_i, n_j)) = \text{Known_Hypernym}$, if

- n_j is an ancestor of the first sense of n_i in the WordNet

$Y((n_i, n_j)) = \text{Known_NonHypernym}$, if

- n_i and n_j are contained in the WordNet
- Neither one is an ancestor of the other in the WordNet for any senses.

If (n_i, n_j) does not satisfy either of the conditions set above, we consider it Unkown and will not include it. In our known set, the portion of Known_Hperynym and Known_NonHypernym are around 6% and 94% of all known pairs.

To evaluate our classifiers, we perform cross-validation on the known set, using a portion to train and a portion to test.

5 Results

We train several classifiers on two different sets of data: dictionary word definitions pairs preprocessed into free text sentences (Section 2) and dictionary definitions injected with a priori knowledge of the definition structure (Section 3). The classifiers include: multinomial logistic regression with ridge estimators [4], multinomial Naïve Bayes, complement Naïve Bayes [5], and the SMO [6]. The classifiers are evaluated using cross-validation and the best F-scores were obtained by multinomial logistic regression with ridge estimators. Our results are summarized in figure 3.

Notably, treating dictionary data as free text results in a very poor F-score. While the precision is very high, its recall is negligibly low. This means that the patterns extracted contributes very little information for deciding hypernymy. The classifier essentially predicts the most likely label according to the prior.

Injecting the dataset with a priori knowledge of the text structure results in much higher F-scores. The best F-score obtained in the case of treating dictionary as free text is 0.006 compared to the best F-score of 0.11 obtained on the dataset injected with structure knowledge.

5.1 Comparison with previous work

Our best results obtained are inferior the basic hypernym classifier obtained by Snow, Jurafsky, and Ng [2]. At the same precision of 45% as obtained by our logistic regression classifier, their best hypernym-only classifier obtained a recall of approximately 28% compared to approximately 15% obtained here. Our results here are nonetheless still respectable considering the much smaller dataset used.

1 The corpus contains noun definitions from the online Merriam-Webster dictionary.

2 We access WordNet 2.1 using MIT Java WordNet Interface 1.1.3 as our WordNet query tool.

6 Conclusions

Our experiments show that taking advantage of a priori knowledge of the text structure in structured natural language texts vastly improves the resulting hypernym pair classifier. By injecting knowledge of the structure, we obtain a respectable hypernym pair classifier using a relatively small corpus.

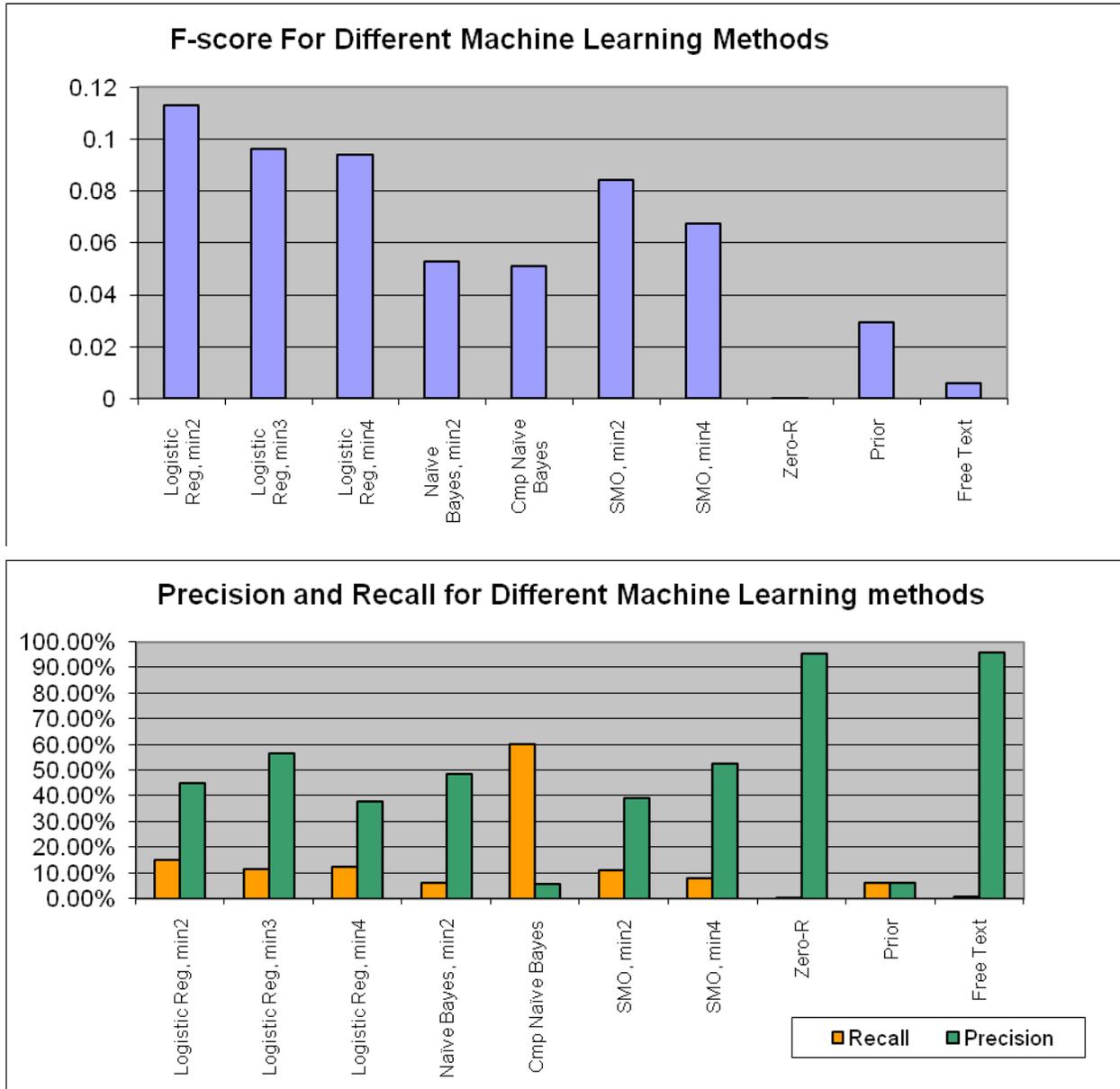


Figure 4: min4 means only attributes occurring in at least 4 distinct word pairs are considered. Zero-R: predicts the most common label, slightly perturbed to produce a valid F-score. Prior: predicts randomly using prior distribution, disregarding attributes. Free text: best result obtained treating dictionary as free text.

7 Future work

7.1 Immediate continuation of this work

Due to machine time and memory constraints, our best classifier (multinomial logistic regression) was trained on a training set that is less than one-sixth of the full training set. So we expect better results when we have the opportunity to use the full training set.

We also lacked the resources to perform wrapper feature selection. When we have the opportunity to perform systematic wrapper feature selection, we anticipate eliminating the currently observed overfitting and improve results significantly.

We also intend to train our classifiers to appropriately trade off precision and recall in order to maximize the resulting F-score.

7.2 Longer term future directions

In this project we merely scratch the surface in using a priori knowledge about the structure of a corpus to improve classifier performance. In the Merriam-Webster dictionary definitions alone, there are many avenues for future investigation. There is often a hierarchy of several definitions given for the same word. In this project, we treat each definition completely independently, but a promising future direction is to leverage interrelations between these parallel definitions using knowledge of the hierarchical structure. Furthermore, there are many other sources of structured natural language text for further exploration, including encyclopedia entries, design documents and legal documents.

Acknowledgments

We thank Daniel Jurafsky and Rion Snow for advising us on this project. They gave us the original idea of investigating semantic taxonomy acquisition in semi-structured text and provided guidance in adapting the existing techniques to semi-structured text to take advantage of the added structure. We thank them especially for their support and encouragement throughout this project.

References

- [1] Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Semantic Taxonomy Induction from Heterogenous Evidence. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics*. 2006.
- [2] Rion Snow, Daniel Jurafsky, and Andrew Ng. Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems 17*. 2005.
- [3] Ian H. Witten and Eibe Frank (2005) "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005.
- [4] le Cessie, S. and van Houwelingen, J.C. (1992). *Ridge Estimators in Logistic Regression*. Applied Statistics, Vol. 41, No. 1, pp. 191-201
- [5] Rennie J., Shih, L., Teevan, J., & Karger, D. (2003) Tackling the Poor Assumptions of Naïve Bayes Text Classifiers. *Proc. Of ICLM-2003*.
- [6] J. Platt (1998). *Fast Training of Support Vector Machines using Sequential Minimal Optimization*. Advances in Kernel Methods - Support Vector Learning, B. Schoelkopf, C. Burges, and A. Smola, eds., MIT Press.