CS 224N: Final Project

Semantic Role Labeling on Nouns using NomBank

Shivaram Lingamneni and Jason Turner-Maier

Background

Treebanks are annotated corpora containing information about syntactical and semantic structure. They are important in the field of natural-language understanding because they allow the application of supervised machine learning algorithms to learning said syntactical and semantic structure. One of the most widely used and annotated corpora is a large set of Wall Street Journal articles. The Penn Treebank annotated this corpus with parse trees; Kingsbury and Palmer created the PropBank project as an extension to this by adding semantic role labels for verbs. For each verb in the corpus, PropBank contains information about senses ("frames") and the arguments typically associated with each sense. Then, annotations to the corpus label verbs and their arguments according to this predetermined scheme.

NomBank is in a sense a twin project to PropBank; it uses the same corpus, instead labeling the arguments of nouns, together with similar word-sense disambiguation. Since NomBank's annotations are similar in both structure and intent to PropBank's, it makes sense to try to adapt existing machine learning techniques for PropBank to NomBank.

David Vickrey, a graduate student in Daphne Koller's AI group at Stanford, has worked on a semantic role labeling system that applies SVM classification techniques to PropBank. We set out to adapt his code to NomBank, producing a semantic role labeling system for nouns; we also decided to investigate relationships between the two datasets. Due to the structure of the semantic information included with NomBank and PropBank, several interesting applications are possible; one example is the rephrasing of sentences containing verb nominalizations (nouns like “assassination” that encapsulate the meaning of a verb) to use the original verb instead.

Algorithms

The code constructs two classifiers. The first identifies words that definitely aren’t frames (nouns that can be labeled), versus words that might be frames. The second classifies frames and their arguments. The first classifier is not strictly necessary, but it may be better at picking out the words which are obviously not frames. Also, the second classifier is multidimensional, so it takes a proportionally longer time to run than the first. Therefore using the first classifier to weed out words to speed up the second classifier saves a lot of time.

The two classifiers are Support Vector Machines. The basic idea behind Support Vector Machines is very similar to other classifiers. In the linear case, you have two types of points in multidimensional space. You want to find a hyperplane that divides them perfectly with all of one type on one side and all of the other on the other side, and you want the distance from the hyperplane to the closest point of each type to be as great as possible. It turns out that this plane is defined by a particular type of vector called a support vector; these are the vectors closest to the hyperplane. It is possible to find the appropriate vectors by means of a minimization of their magnitude subject to a particular constraint, and this gives you your classifier.

The multidimensional case is a bit more complicated. Here you have more than two types of points you would like to classify. The surface that you would like to use to divide them would need to be more complex, and your previous algorithm would fail to work. However, it turns out that there is a method called the "kernel trick" to formulate the multidimensional classifier in such a way that the original algorithm would work by mapping the space into a different set of dimensions.

The classifiers use a number of features to determine if a word is of a particular type. It uses a data-rich representation of the sentence to derive these features, many of which are dependent on the sentence parse as a whole. There is a standard set of features which were being used in the classifier when it was given to us by David Vickrey. They include the target word, the parse path to the target word, whether the current word is before/after the target word, the head word of the phrase, and so on. There are also pairwise features which combine two or more other features, for example the target word and the head word. They are used when two pieces of information are thought to correlate in some way, or to provide additional information together from when they were separate.

This two-pass system, together with the use of parse information in the features, appears in the literature (Jurafsky) as a current and effective method of semantic role labeling.

Adapting the System

Although NomBank is syntactically similar to PropBank, it required an extended process of text pre-processing, together with small adjustments to the code, before we could run it successfully. Some of the issues required compromises. For example, NomBank uses some tags that are not in the PropBank specification in order to assign different semantic roles to the different parts of a hyphenated word or phrase. Since we could not realistically adapt the existing code to make use of this information, we simply dropped the tags. One possible area for future investigation (short of adding full support for hyphenation) is collapsing the hyphen tags, overloading the hyphenated word with all its semantic roles.

Once we had it running, it was fairly straightforward to adapt the classification and evaluation procedures to the new data set. The existing system had a very comprehensive range of features. However, we saw an opportunity for extension in the NomLex file that comes packaged with NomBank. NomLex is effectively a dictionary rather than a corpus, containing a list of properties of nouns, together with some information about their potential semantic roles. One datum given for almost every noun is a "NOM-TYPE," which reflects whether a noun is a nominalization and what kind of nominalization it is. For example, "casualty" is an "OBJECT", while "caroler" is a "SUBJECT", "concussion" is a "VERB-NOM" (verb nominalization), and "confidentiality" is an "ADJ-NOM". We decided to use these types as features to the classifier, pairing them with the parse features to create polynomial features.

One problem with our use of NomLex is that it was not integrated with the system of frames and word-sense disambiguation used by NomBank and PropBank. For example, it lists "acrimony" as both an "ADJ-NOM" (in the sense of "acrimoniousness") and a "VERB-NOM" (in the sense of a conflict or clash); no cross-reference to a possible frame-based distinction in NomBank is given. Since we could only associate one type with the ambiguous noun name, we simply chose the last one given. One area for further investigation might be a better way to choose the type in these cases, although the problem is sufficiently rare that no significant benefits might be derived.

Results

Diagnostic: Nombank on 50 frames, with the noun type crossed with the listed feature

Feature	Precision	Recall
(none)	0.900585	0.785714
ambiguous_frame_name	0.900585	0.785714
head_word	0.894737	0.780612
path	0.928144	0.790816
phrase_type	0.906433	0.790816
path_length_string	0.916667	0.785714
head_word_phrase_type	0.905325	0.780612
parent_phrase_type	0.911765	0.790816
first_word	0.901163	0.790816
last_word	0.905325	0.780612
is_passive	0.900585	0.785714
left_phrase_type	0.905325	0.780612
right_phrase_type	0.900585	0.785714
left_head_word	0.912281	0.795918
right_head_word	0.883721	0.77551

Effect of the number of frames on precision:

Number of Frames	Propbank	Nombank (base)	Nombank (augmented)
10	0.930857	0.963504	0.942029
25	0.929021	0.971223	0.943262
50	0.923655	0.900585	0.928994
100	0.909869	0.907631	0.858238

Effect of the number of frames on recall:

Number of Frames	Probank	Nombank (base)	Nombank (augmented)
10	0.91947	0.88	0.866667
25	0.919929	0.89404	0.880795
50	0.904966	0.785714	0.80102
100	0.8924	0.81295	0.805755

Number of correctly labeled sentences for the base and augmented classifiers over different numbers of frames:

Number of Frames	Nombank (base)	Nombank (augmented)
10	132	130
25	135	133
50	154	157
100	226	224

Number of incorrectly labeled sentences for the base and augmented classifiers over different numbers of frames:

Number of Frames	Nombank (base)	Nombank (augmented)
10	5	8
25	4	8
50	17	12
100	23	37

Analysis

As seen in the table above, we found that the most effective features to pair with noun type were path, path length, phrase type, parent phrase type, first word, and left head word. We therefore added all of these features to the classifier. On 50 frames, this seemed to be efficacious, adding about 2% to the precision. However, on 100 frames, these additional features actually dropped the precision by about 5%. There are two probable explanations for this.

First, it is possible that which features were helpful on the first 50 frames was not representative of the next 50 frames. Most of the errors were caused by the classifier labeling words that shouldn't be labeled, so it is possible that the extra features weren't adding much in the way of actual information and were just making the classifier more likely to label any given word.

Second, it is possible that the additional features caused the classifier to overfit to the training data. Including additional features means that more information is being stored about the training set examples, in effect, and if that information is not particularly useful in determining the relevant classifications it will just cause the classifier to pick words that are similar to the words in the training set.

An interesting feature of the results can be seen when the number of correct sentence assignments for the base nombank classifier versus the number correct for the augmented nombank classifier is graphed, and the same is done for incorrect sentence assignments (please see the excel file for these graphs). The number correct matches up fairly well between the two classifiers. However, the number incorrect varies proportionally more widely. This suggests that while both classifiers label correctly with the same degree of accuracy, their principal difference is in how many they label incorrectly/fail to label correctly. This lends more credence to the supposition that the performance of these features on 50 frames was not representative of their performance overall, because the graphs indicate that of 10, 25, 50, and 100 frames, the only time that the augmented classifier labels more sentences incorrect than the base classifier is for 50 frames.

In terms of what errors are made most commonly, an examination of the error logs reveals that most specific mislabelings for both classifiers were the labeling of words that weren't arguments as arguments, or vice versa; it was rare that the classifier mislabeled a word that is an argument as the wrong argument type. Therefore, the classifier is very good at determining the correct argument type once it has found an argument, but less good at identifying the specific arguments. A good avenue of further research then might be to train a classifier that just tries to label words as arguments or not arguments and lets a later stage determine which arguments they are.

Application: Rephrasing

One of the questions we began with was the problem of taking a sentence containing a nominalization and rephrasing it to use the corresponding verb. In addition to its labeling of noun arguments within the corpus, NomBank contains a considerable amount of syntactic and semantic information. NomBank's frame files contain, in XML format, a brief description of the meaning of the arguments of every word (in every sense) labeled in the corpus, together with XML-tagged sample sentences. Furthermore, NomBank is packaged with (and in some sense built on top of) the NomLex project mentioned earlier, which seeks to classify nouns by type and associate them with the syntactic properties of their arguments. Given the amount of available data, we realized that rephrasing sentences would be simple in principle but difficult to put into practice. What follows is a description of how such a system could be implemented.

Our classifier can label the arguments of an arbitrary phrase in the corpus, for example: "The assassination of President Kennedy by Lee Harvey Oswald." (The requirement that the phrase occur in the corpus has to do with the fact that we only have sufficiently detailed semantic information for words that occur in the corpus.) The classifier gives us that "assassination" occurs in its sense of "assassination.01", and that Kennedy was ARG1 and Oswald ARG0. Now, we can consult the NomBank frame file to determine that "assassination.01" is associated with the verb "assassinate.01" in PropBank, together with the fact that Kennedy was the patient (ARG1) and Oswald the agent (ARG0) of the assassination. Now, consulting the PropBank frame file for "assassinate.01", we can choose between an active and a passive construction. In the first case, the XML tags indicate to put ARG0 in front, the verb in the middle, and ARG1 at the end; in the second, the order is reversed. We might end by producing "Lee Harvey Oswald [assassinate] Kennedy."

The significant remaining issue is conjugating the verb. In this, we are helped by the fact that our classifier does a machine parse of the sentence; in particular, it can access the person and number of each argument, together with the tenses of any verbs. In fact, whether the sentence is active or passive is already available as a feature of the parse, so acquiring this additional information should not be difficult.

It is possible to replace certain uses of the frame files in this process with references to NomLex, which contains more data and would work on sentences with words not occurring in the corpus; however, this comes at a cost. NomLex is far more liberal in its association of nouns with verbs. For example, it considers "carnival" to be a nominalization of "commemorate." Second, it does not include word-sense disambiguation, creating a variety of possible problems with heavily overloaded nouns.

Topics for further investigation

A number of possible extensions have been mentioned throughout the paper. However, one item in particular deserves mention. In the middle of our project, a new version of NomBank was released; the new version includes more comprehensive annotations, together with cross-links between compound/plural nouns and their constituent parts. It is possible that this information could simplify the implementation of rephrasing or improve the precision of the labeling system.

Acknowledgements

We would like to thank David Vickrey for his help with the project.

Works Cited

Jurafsky, Dan et al. "Parsing Arguments of Nominalizations in English and Chinese."