Frequency-Based Measures for Predicting Syntactic Relationships

Simon Fung
University of Alberta
Edmonton, AB
simonmin@ualberta.ca

1 Introduction

In unsupervised syntactic parsing, an algorithm receives as input either unlabeled text documents or, more often, sequences of part-of-speech (POS) tags corresponding to the words in the documents. From this data, it infers the syntactic structure of each sentence, usually either in the form of constituents or dependency relations. Three such parsers that are the basis of my investigation are CCM (Constituent-Context Model) and DMV (Dependency Model with Valence) from Klein (2004), and U-DOP (Unsupervised Data-Oriented Parsing) from Bod (2009). DMV produces dependency trees, while CCM and U-DOP produce constituency trees. All three algorithms ultimately use cooccurences and conditional probabilities of POS tag pairs in a corpus.

In this study, I look at how well cooccurence rates and conditional probabilities could be used to differentiate syntactically-related POS tag pairs from unrelated ones. I compare the average cooccurence rates and conditional probabilities of POS tag pairs that are syntactically-related in a labeled corpus, to those of tag pairs that are not related. In addition, I also look at the effect of corpus size on the measures, the separability of related pairs from unrelated ones, and the correlation between the measures and the likelihood of a POS pair to be syntactically-related.

Since Klein (2004, 2005) and Bod (2009) both evaluated their system on treebanks in English, German, and Chinese, I chose treebanks in Arabic and Japanese, two other languages that are genetically and typologically quite different. I used dependency treebanks for both languages, since dependency trees are more informative and more flexible than constituency trees.

2 Background

CCM and U-DOP, the two constituency parsers, are quite different in their approaches: the former is an Expectation-Maximization (EM) algorithm that finds the probability distribution of trees that maximizes the likelihood of the occurence of the observed sentences, while the latter produces the structure that is as similar as possible to the sentences it has already seen. However, both appear to rely on cooccurence. On the other hand, DMV depends on the conditional probability of generating a dependent word given the head word.

2.1 U-DOP

For all the sentences in the training data, U-DOP stores all possible trees that could cover them, and stores all the subtrees as well. It then considers the best constituent tree for a sentence to be the "Most Probable tree generated by the Shortest Derivation ('MPSD') of the sentence". With the subtrees in its storage, U-DOP finds the derivation for a given sentence using the least number of derivations. Where there are ties, the most probable tree is chosen. The probability of a tree is defined as the sum of the probabilities of all its possible derivations, and the probability of a derivation is defined as the product of the probabilities of all the subtrees involved.

A sequence is more likely to be identified as a constituent in U-DOP if they occur together more often. For example, assume the algorithm sees two training sentences, *Watch the dog* and *The dog barks*. Among the possible binary trees for those sentences, the only constituent that occurs more than once is *the dog*. A high frequency of cooccurence increases the probabilities of subtrees that contain them, thereby making it more likely that the consequence will appear as a constituent in the winning tree.

2.2 CCM

As in U-DOP, the underlying structure in CCM is a binary constituent tree structure. The model's parameters consist of two conditional probabilities. P(yield|constituency) is the probability of various POS sequences (the *yield*) occurring, given that we know whether or not they form a constituent; and P(context|constituency) is the probability of various POS pairs (the *context*) bookending a span, given that we know whether or not that span is a consituent. For each sentence, the algorithm first fixes these parameters and finds P(tree|sentence), the probability distribution of possible trees given a sentence. Then, it fixes the probability distribution of possible trees, uses it to calculate over all possible trees the expected value of the probability of generating the sentence, and finds the new parameters that would maximize this expected value. These two steps alternate, until convergence is reached.

Essentially, CCM gives high probabilities to the trees whose constituents have a high probability of being constituents, and gives high probabilities of being constituents to the POS sequences that often occur as constituents in the trees that could possibly cover them. Therefore, in a way similar to U-DOP, the POS sequences most likely to be constituents are the ones that occur together most often in the same sentence.

2.3 DMV

In DMV, another EM algorithm, sentences can be thought of as being generated one by one, starting from a special ROOT node, which generates one child, the word that is the actual root of the dependency tree. Subtrees are then generated recursively from dependents. The algorithm first fixes the parameters to calculate the probabilities of various trees for a given sentence. Then, the algorithm takes the probabilities of the various dependency trees and finds the parameters that maximize the probability of generating the observed sentences. The parameters are the probability of stopping the generation of dependents from a given head, and the probability of generating various words (again, delexicalized and replaced by their POS tags). These are all conditional probabilities; in particular, the probability of generating various words is dependent on what the head word is, and the direction of the dependency (left or right). Therefore, the pairs of POS tags most likely to be assigned a dependency relation by the algorithm are those whose dependents are most likely to occur given that the head has occurred in the sentence. If the dependent word has a high probability of occurring given that the head word is present, trees that connect them will receive a higher probability, and in turn the probability of generating the dependent given the head would increase to maximize the probability of generating the observed sentences.

2.4 Dependency vs. Constituency Trees

In this project, I chose to work with dependency trees instead of constituency trees. One advantage of dependency trees is that they show syntactic relations more explictly. They are more flexible, in that subtrees are not limited to covering contiguous substrings, which is a stipulation in most uses of constituency trees. This property, known as projectivity, does not always hold in natural language (Mel'čuk 1988). Dependency trees are also more informative than constituency trees; besides containing all the information available in constituency trees (all dependency subtrees form a constituent), dependency trees also contain headedness information, as well as the directions of syntactic relationships. Although phrase labels like VP can indicate which of its constituents is the head, constituent class labeling is not generally part of unsupervised parsing. Klein (2005) does later include constituent categorization in CCM, with mixed results. However, after categorization, the algorithm must then identify the heads of the members in each class, a non-trivial task, whereas this information is already inherent in a dependency structure.

3 Methodology & Results

3.1 Corpora

I used two dependency treebanks: the Prague Arabic Dependency Treebank (54,379 tokens) and the Japanese Verbmobil treebank (151,461 tokens). These were both used in the CoNLL-X Shared Task on Multi-lingual Dependency Parsing. Since the Arabic treebank consists of written text while the Japanese treebank consists of spoken data, the sentences in the Japanese treebank tended to be shorter. Consequently, the total number of same-sentence word pairs (explained below) was 1,443,711 for Arabic, and 1,263,536 for Japanese.

Each corpus was tagged for both coarse and fine parts of speech, with the former tagset being smaller than the latter. The sizes of the Arabic tagsets were 15 for the coarse and 20 for the fine;

for Japanese, they were 22 for the coarse and 83 for the fine. As in Klein (2004, 2005) and Bod (2009), I used these POS tags, both to avoid sparseness and to focus on syntactic relations rather than semantic ones.

3.2 Setup

I counted the cooccurences of POS pairs in the same sentence, keeping a separate count for each pairing (e.g. N and V). For the purposes of this project, a cooccurence represents a potential syntactic dependency. Therefore, instead of counting just once when both POSs occur in the same sentence, I count how many different ways the pair could be formed in the sentence; for example, in a sentence consisting of three Ns, there would be three cooccurences: the first two Ns, the first and last, and the last two.

Three measures were then calculated for each POS pair: the cooccurence rate, and the conditional probability of the second POS given the occurence of the first one, and the probability of the first POS given the occurence of the second. To calculate the *cooccurence rate*, the total raw cooccurence count for each POS pair was divided by the maximum number of cooccurences possible in the corpus for that POS pair. For heterogeneous POS pairs, this occurs when each POS occupies as close to half the sentence as possible: $n^2/4$ for sentences with an even number of words, and *floor(n/2) * ceil(n/2)* for those with an odd number of words, where *n* is the number of words in the sentence. For pairs consisting of two identical POSs, the maximum occurs when the entire sentence is made up of that POS: *n(n-1)/2*.

For each corpus and tagset, the POS pairs were divided into two sets: those that were syntactically-related at least once in the corpus, the *dependency set*; and those that were never syntactically-related, the *non-dependency set*.

3.3 Part 1: Comparison of Averages

In the first step, averages of each of the three measures mentioned above were calculated for each set. Table 1 shows the results for the four corpus-tagset combinations. All scores were considerably higher in the dependency set than in the no-dependency set. Ratios of average conditional probabilities in both directions for the dependency set to the non-dependency set ranged from 1.80 to 3.00. The ratios for average cooccurence rates were even higher, ranging from 4.58 for the Japanese corpus with the fine POS tagset, to 186.24 for the Arabic corpus with the coarse tagset.

Table 1: Average cooccurence rates and conditional probabilities

	Average Cooccurence Rate	**Average P(arg|head)**	**Average P(head|arg)**	**P(arg|head) / P(head|arg)**
Arabic, Coarse POS				
Dep	0.0241	0.0848	0.0877	0.9675
No-dep	0.000129	0.0376	0.0312	1.2081
Dep/ No-dep	186.2400	2.2530	2.8133	--
Arabic, Fine POS				
Dep	0.0145	0.0666	0.0693	0.9606
No-dep	0.000509	0.0278	0.0231	1.2042
Dep/ No-dep	28.5468	2.3923	2.9990	--
Japanese, Coarse POS				
Dep	0.0117	0.0579	0.0621	0.9336
No-dep	0.00176	0.0321	0.0260	1.2336
Dep/ No-dep	6.6390	1.8033	2.3829	--
Japanese, Fine POS				
Dep	0.00157	0.0238	0.0231	1.0321
No-dep	0.000342	0.01166	0.0120	0.9732
Dep/ No-dep	4.5785	2.0451	1.9284	--

For every corpus-tagset combination except for Japanese-fine POS, the ratio for P(head|arg) is somewhat higher than that of P(arg|head). Interestingly, the ratios between average values of P(arg|head) and P(head|arg) do not appear significant. This suggests that headedness does not correlate strongly with either conditional probability.

Ideally, the significance of these ratios would be quantified with p-values; however, calculating p-values in this case is non-trivial, due to constraint of the dependency tree structure. Instead, significance can be inferred from the consistency of these results across the various corpus-tagset combinations.

The variation in average cooccurence rates seems to correlate with both the size of the tagset and the number of word pairs in the corpus. The largest ratio in cooccurence rates between the dependency and no-dependency sets is in the larger Arabic corpus (in terms of total POS pairs) with the coarse tagset, while the smallest difference ratio is in the Japanese corpus with the fine tagset. This suggests that having more word pairs in the corpus can help to differentiate de-

pendency POS pairs by cooccurence rate, a possibility that I investigated next.

3.4 Part 2: Effect of Corpus Size

I next repeated the process in the previous section for each corpus-tagset combination, for the initial portion of the corpus, increasing this portion one sentence at a time, from consisting of just the first sentence to the entire corpus. The results are shown in Figures 1-6.

Cooccurence rates show an overall increase with corpus size, although there are temporary decreases as well; however, all four rates appear to level off towards the end. Conversely, conditional probabilities do not seem to increase significantly with corpus size; rather, the trend is to increase quickly for smaller sizes, then decrease to a lower value, where it then hovers. The only corpus-tagset combination without a significant decrease is Japanese-fine POS tagset. It is also the combination which does not have a curious sudden sharp increase towards the middle of the graph for P(head|arg); the cause and significance of this is unknown.

3.5 Part 3: Breakdown by POS pairs

Having seen a consistent difference in all three measures between the dependency set and the non-dependency set, I wanted to know how separable the two sets are by all three measures, how much they overlap. Figures 7-18 show the scatter plots of the two sets for all measures in all corpus-tagset combinations. Figure 19 shows the data in Figure 7 on a log scale.

Table 2 shows the percentage of POS pairs that lie within the overlap for each corpus-tagset combination. Overall, the two sets overlap quite substantially; the differences in their averages seem to be due to the POS pairs in the non-dependency set clustering at the lower end of the range encompassed by the dependency set. However, the corpus-tagset combination that has the least overlap, Arabic-coarse POS, is also the one with more POS pair cooccurences and the smaller tagset. It is possible that with larger corpora and the same number of POS tags, the amount of overlap will decrease. Notice that the overlaps for the conditional probabilities are quite high even for Arabic-coarse POS.

Table 2: Percentage of POS pairs in overlap between dependency and non-dependency sets

	Arabic		Japanese	
	Coarse	Fine	Coarse	Fine
Cooccurence	30.5164	81.1518	86.8251	91.7178
P(arg|head)	93.8967	96.5969	93.3045	98.3896
P(head|arg)	93.4272	95.2880	92.8726	99.7316

3.6 Part 4: Correlation between cooccurence rate and frequency of depencency

The last investigation looked at whether POS pairs with higher cooccurence rates tend to have higher probabilities of being syntactically dependent where they do occur. To do this, I calculate the correlation between each of the three measures (cooccurence rate and conditional probabilities in both directions) and the proportion of cooccurences of each POS pair that are in fact syntactically-related. Table 3 shows the results.

Table 3: Correlations of POS pairs between 3 measures and probability of dependency given cooccurence

	Arabic		Japanese	
	Coarse	Fine	Coarse	Fine
Cooccurence	-0.0792	-0.0556	-0.0436	-0.1242
P(arg|head)	-0.1240	-0.0941	0.1030	0.0464
P(head|arg)	-0.0884	-0.0569	-0.0802	-0.0654

It turns out that none of the measures are very highly correlated with the probability of an cooccurence being syntactically-related. In fact, more often than not, they are negatively correlated. This means that while cooccurence rates and conditional probabilities of a POS pair are correlated with whether or not the pair is ever syntactically-related, they do not predict how often the pair is a dependency when it does occur.

4 Conclusions

The results show that on average, cooccurence rates and conditional probabilities in both directions for syntactically-related POS pairs are considerably higher than those for unrelated pairs, with average cooccurence rates being especially differentiated. This difference increases with corpus size, although the effect appears to level off gradually. When broken down by individual POS pairs, the syntactically-related set and the unrelated set show substantial regions of overlap

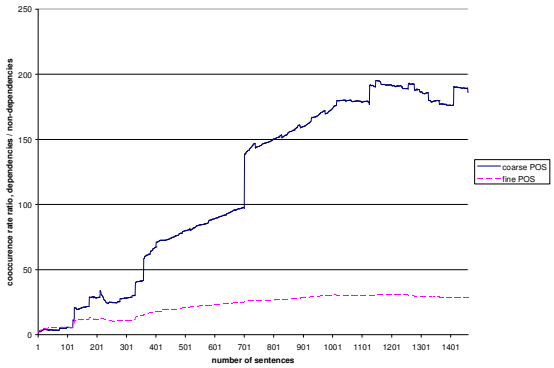

Figure 1: Cooccurence rate ratio of dependencies/non-dependencies, Arabic

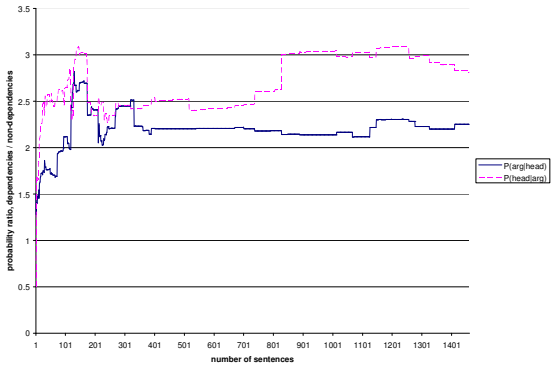

Figure 2: Conditional probability ratio of dependencies/non-dependencies, Arabic, coarse POS

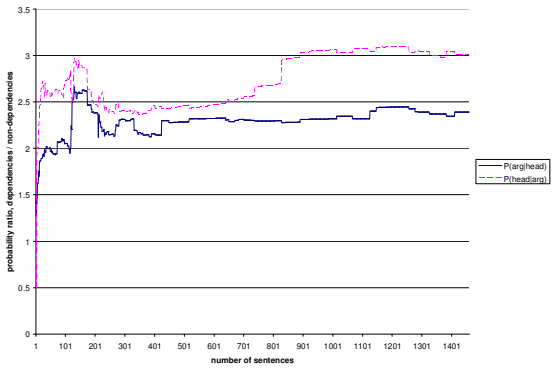

Figure 3: Conditional probability ratio, dependencies/non-dependencies, Arabic, fine POS

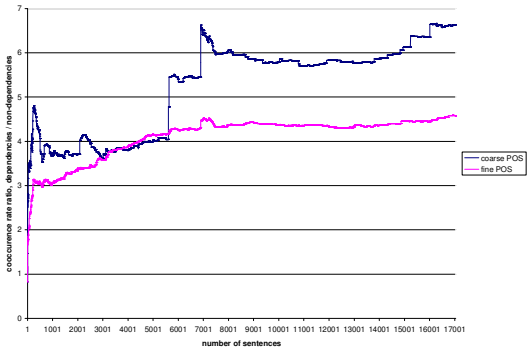

Figure 4: Cooccurence ratio of dependencies/non-dependencies, Japanese

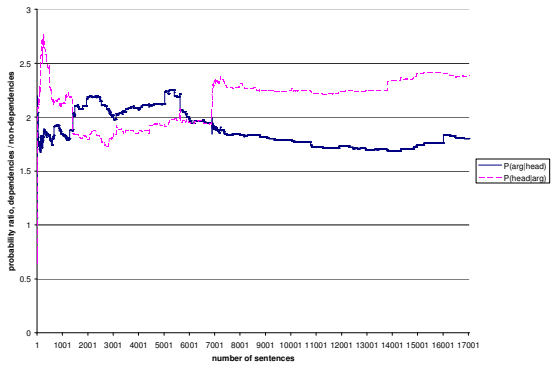

Figure 5: Conditional probability ratio, dependencies/non-dependencies, Japanese, coarse POS

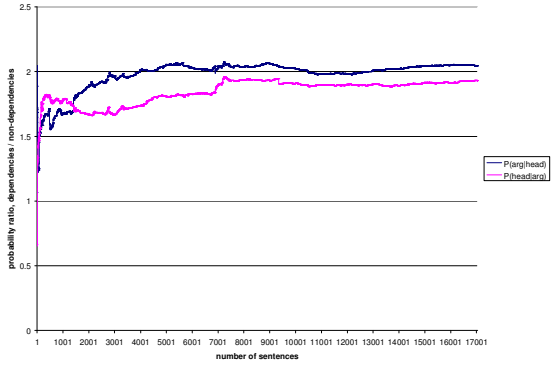

Figure 6: Conditional probability ratio, dependencies/non-dependencies, Japanese, fine POS

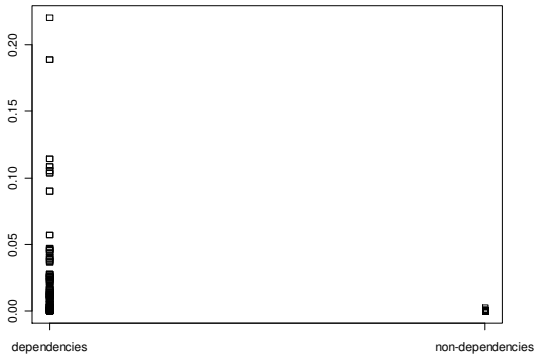

Figure 7: Cooccurence rates for specific POS pairs, Arabic, coarse POS

Figure 8: Cooccurence rates for specific POS pairs, Arabic, fine POS

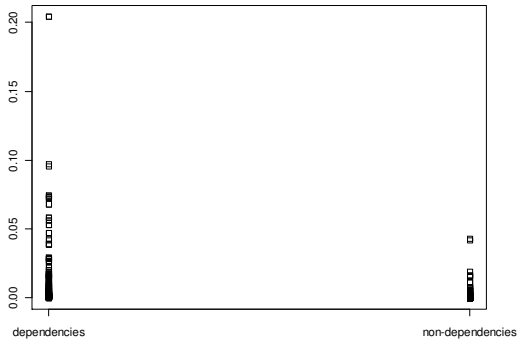

Figure 9: Cooccurence rates for specific POS pairs, Japanese, coarse POS

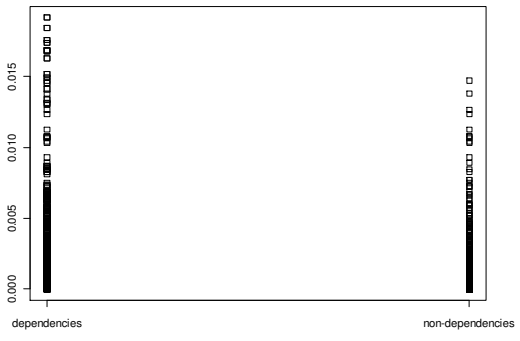

Figure 10: Cooccurence rates for specific POS pairs, Japanese, fine POS

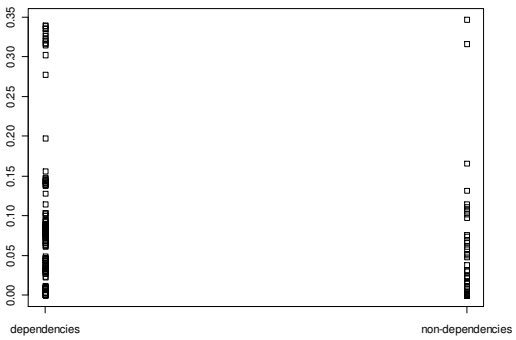

Figure 11: Ratio of P(arg|head) for dependencies/non-dependencies, Arabic, coarse POS

Figure 12: Ratio of P(arg|head) for dependencies/non-dependencies, Arabic, fine POS

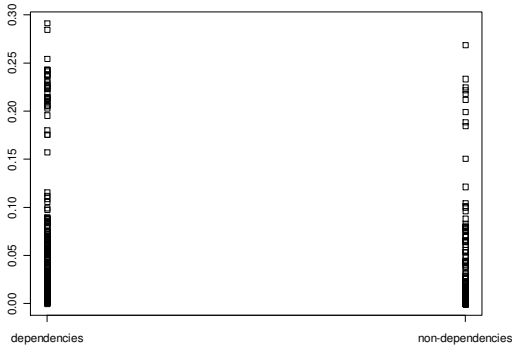

Figure 13: Ratio of P(arg|head) for dependencies/non-dependencies, Japanese, coarse POS

Figure 14: Ratio of P(arg|head) for dependencies/non-dependencies, Japanese, fine POS

Figure 15: Ratio of P(head|arg) for dependencies/non-dependencies, Arabic, coarse POS

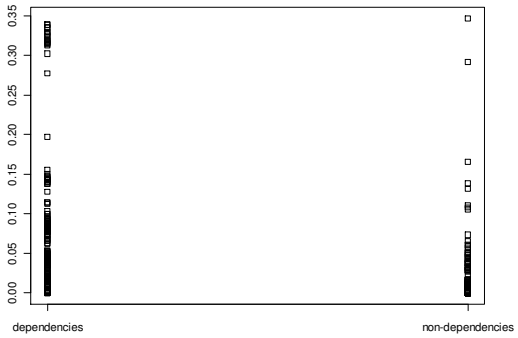

Figure 16: Ratio of P(head|arg) for dependencies/non-dependencies, Arabic, fine POS

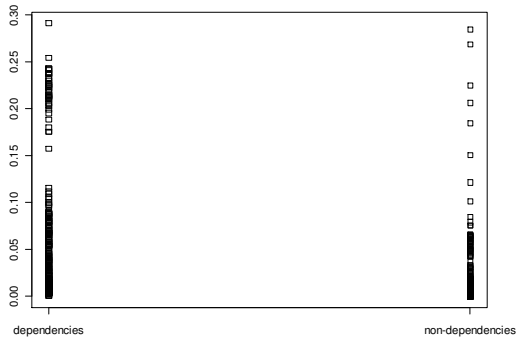

Figure 17: Ratio of P(head|arg) for dependencies/non-dependencies, Japanese, coarse POS

Figure 18: Ratio of P(head|arg) for dependencies/non-dependencies, Japanese, fine POS

Figure 19: Cooccurence rate ratio of dependencies/non-dependencies, log scale, Arabic, coarse POS

on all three measures; however, the corpus with more cooccurences and fewer POS tags, Arabic-coarse POS, has a much lower overlap on cooccurence rates, suggesting that a larger corpus relative to the number of tags may show less overlap. A further caution is that while POS pairs that are syntactically-related at least once in the corpus have higher cooccurence rates and conditional probabilities on average, higher values do not correlate with being syntactically dependent more often, when a POS pair does cooccur.

Overall, cooccurence rate appears to be better at differentiating between syntactically-related and unrelated pairs than either conditional probabilities. Moreover, there is no substantial difference between P(arg|head) and P(head|arg), meaning that conditional probabilities cannot indicate headedness in a syntactically-related POS pair.

Cooccurence rates were the basis for the two consituency models, CCM and U-DOP, while the conditional probability P(arg|head) was the basis for the dependency model, DMV. This study strongly suggests that cooccurence rate is better at distinguishing dependency model as well, compared to conditional probabilities.

5 Future Work

Since significance tests are difficult to apply to this study, confirmation from repeated tests are important. In particular, I would like to run similar tests on a much larger corpus, to see if that will exaggerate cooccurence rates between syntactically-related and unrelated POS pairs. One possibility is to run a supervised dependency parser, such as the Stanford Parser (Klein & Manning 2003), on a large unlabeled English corpus. This can also be done for Japanese, with CaboCha (Kudo & Matsumoto 2003), or any other language with a reasonably good supervised dependency parser.

This work is part of my preliminary work towards achieving better results in unsupervised dependency parsing. The challenge now is figuring out how to use cooccurence rates to find the best dependency tree.

References

Bod, R. (2009). From Exemplar to Grammar: A Probabilistic Analogy-based Model of Language Learning. *Cognitive Science*, 33(5), 752-793.

Klein, D. and Manning, C. (2003). Fast Exact Inference with a Factored Model for Natural Language Parsing. In *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, Cambridge, MA: MIT Press, 3-10.

Klein, D. (2004). Corpus-Based Induction of Syntactic Structure: Models of Dependency and Constituency. In *Proceedings of the Association for Computational Linguistics (ACL) 2004*.

Klein, D. (2005). *The Unsupervised Learning of Natural Language Structure*. Ph.D. Thesis, Stanford University.

Kudo, T. and Matsumoto, Y. (2003). Fast Methods for Kernel-Based Text Analysis. In *Proceedings of the Association for Computational Linguistics (ACL) 2003*.

Mel'čuk, I. (1988). *Dependency Syntax: Theory and Practice*. Albany: SUNY Press.