

Unsupervised Probabilistic Grammar Induction

Benoit Essiambre

School of Information Technology and Engineering
University of Ottawa
bessiamb@site.uottawa.ca

Nathalie Japkowicz

School of Information Technology and Engineering
University of Ottawa
nat@site.uottawa.ca

Abstract

In this research we combine bootstrapping and clustering in order to implement a probabilistic hidden structure model of language that incrementally learns a probabilistic dependency grammar. We use a clustering algorithm that builds clusters that represent grammatical elements containing lexical and categorical information. The grammar is specified probabilistically through a measure similar to *lexical attraction* which is defined as the likelihood of two words being related. Preliminary results are promising.

1 Introduction

The challenge of unsupervised grammar induction, where a grammar is inferred from example corpora, can not only help us build better natural language parsers but it can also give us insights about the possibilities and limitations of using computers to algorithmically reproduce certain cognitive abilities of the human mind.

Some linguists see the rules of language as categorical and discrete and some think that they are probabilistic and continuous. In recent years there have been vast efforts in using probabilistic models and machine learning on language yet the performance has often not been much higher than in previous systems. (Manning and Schütze, 1999) Nevertheless many researchers still challenge the idea that linguistic competence is categorical and discrete. They assert that it is time to re-evaluate the way we do probabilistic processing of language. (Bod et al., 2003b) Several probabilistic

approaches are failing because they restrict their analysis to that of surface facts rather than try to discover more abstract models and hidden structures that could explain surface facts with more coverage and higher accuracy. (Bod et al., 2003b) Manning (2003) argues that categorical linguistic theories claim too much and explain too little by placing hard categorical boundaries and ignoring soft constraints.

Probabilistic models that make use of abstract hidden structure have the potential to solve these problems. They can integrate word frequency, gradient of categories and morphological productivity. The “argument from poverty of stimulus”, which argues language is largely innate by attempting to prove that it isn’t learnable from examples, has been confronted notably, with evidence that “Unlike categorical grammars, probabilistic grammars are learnable from positive evidence alone” (Bod et al. 2003b).

Systems that make use of complex probabilistic structures to process language have recently been developed to parse context free grammar (Bod et al., 2003a) and learn dependency grammar (Klein and Manning, 2004). In this research we use a novel approach that combines bootstrapping and clustering in order to incrementally learn a probabilistic structure of dependency grammar.

2 Dependency Grammar

Following the Chomskyan tradition, most research in language acquisition is based on phrase structure grammar. Many arguments, however, justify our use of dependency grammar. Dependency grammar makes lexical relations more explicit (Collins, 1999). Dependencies are fitting to machine learning and probabilistic methods because they can be used as lexical features of words or

word categories. Also, the head to word relation is implicit in dependency grammar whereas it has to be specified in phrase structure grammars. Levy and Manning (2004) argue that phrase structure grammar is merely a “safe approximation” to dependency grammars when dealing with non-local dependencies and they go so far as declaring that the phrase structure representation approach might be a “dead end”. Nevertheless the two types of grammars are isomorphic and parsed sentences can be translated from one representation to the other. (Klein and Manning, 2004) This allows research using one type of grammar to be relevant to the other type.

3 Algorithm

Most linguistic features of words exhibit *probabilistic* dependency (as opposed to linguistic dependency) with other features. It is a well known fact that most syntactic features of words are correlated with particular morphological, phonological and semantic features. For example, verbs which often connect syntactically to nouns, tend to be associated with certain morphemes (-ing -ed) and usually convey an existence or an action. Some of this dependency serves as redundancy and can (and should) be used for linguistic disambiguation and learning.

One way to capture the information contained in probabilistic dependency is to use a clustering technique on words and their features. In order to implement a probabilistic hidden structure model of language, we use a clustering algorithm that builds clusters that represent grammatical elements containing lexical and categorical information.

The incremental clustering algorithm was created to enable the possibility of bootstrapping. For each sentence on which the system learns, it is parsed and the output is fed back as extra features of the words. Yuret (1998) has shown that boot-

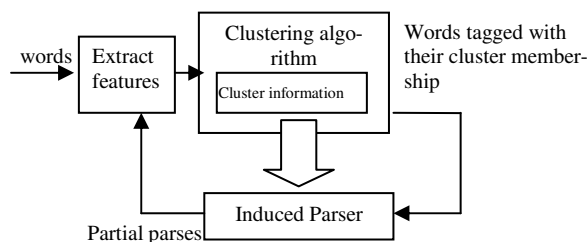


Figure 1. System Overview

strapping can be used to discover deep relationships that are not captured by standard learning techniques. In this research bootstrapping is utilized as a key method to get away from surface probabilistic facts.

Other extracted features are morphemes, word stems, adjacent word classes, word types and first letter cases. The system is trained on plain texts from Project Gutenberg¹.

3.1 Original Parsing Algorithm

The sentences are parsed by using a modified version of a greedy algorithm (Yuret, 1998). The following describes Yuret’s original algorithm. The algorithm searches for the most probable acyclic and planar dependency tree. It has the important benefit of being constructed to work in a bootstrapping framework.

The grammar is specified probabilistically through *lexical attraction* which Yuret defines as the likelihood of two words being related. In the terms of information theory, it is the pointwise mutual information gained by linking these two words through a dependency.

$$MI(x, y) = \log_2 \frac{p(x, y)}{p(x, *)p(*, y)}$$

Since the method is unsupervised, lexical attraction must be learned from a corpus using bootstrapping. In the beginning, the processor counts the occurrence of adjacent words to build word pair frequencies. This allows the parsing algorithm to make partial parses of the sentences it sees. The processor then uses the parsed sentence to add further dependency counts adjacent to current links. In a sequence AX...YB where X and Y are linked by a dependency, the processor would count the links A-Y and X-B in addition to the adjacent word pairs A-X and Y-B. This enhances the learning and allows the model to learn higher level and more abstract word relationships.

3.2 Modified Parsing Algorithm

Our system uses a modified version of Yuret’s algorithm. First, instead of using mutual information we use Information Gain (IG). We can use the fact that IG is computed from the perspective of a particular word to infer a direction to dependencies.

¹ <http://www.gutenberg.org/>

The assumption here is that words tend to have the same kind of head but a variety of dependents.

$$IG_x(x, y) = \log_2 \frac{p(x, y)}{p(x, *)}$$

$$IG_y(x, y) = \log_2 \frac{p(x, y)}{p(*, y)}$$

Second, Yuret's algorithm uses word type as the base unit for the frequency counts and as a result it does not take into account any other feature of words. In an attempt to capture more information, we use our clustering algorithm on the words and translate the concept of lexical attraction to use word clusters instead of word types as our base units. Each word in a sentence is assigned to a cluster and IG is computed for dependency links between clusters. One immediate advantage is that never seen words can still be linked based on their morphological and syntactic features. For example, a never seen word ending in -ed following a pronoun would tend to be clustered with other simple past verbs and linked with the same words. The phenomenon of morphological productivity clearly emerges from the clustering.

3.3 Clustering Algorithm

It is impossible to use an existing clustering algorithm for this task because none are incremental and have the speed we require. Instead we use a distance based clustering algorithm that is straightforward. The processor starts with no clusters. Then each time an instance of a word is encountered that can't, based on the proximity function, be assigned to an existing cluster, a new cluster is created that contains only this instance. Otherwise no cluster is added and the instance is assigned to near clusters.

Clustering while using bootstrapping in an incremental setting leads to a few problems. Systems with feedback loops can be unstable. This fact is quite familiar to those studying dynamic systems. Instability can exhibit itself in different forms, but it usually results in the system parameters either diverging to 0, diverging to \pm infinity or oscillating wildly instead of converging to a constant value. As an example of how this can happen, suppose a distance based clustering algorithm where the proximity function is Bayes rule : $\Pr(X|W) = \Pr(W|X)\Pr(X)/\Pr(W)$ and the prior probability $\Pr(A)$ is calculated using Maximum Likelihood

Estimation (the quotient of the occurrence of instances assigned to cluster $A=i$ and the total number of occurrences: $C(i)/C_T$). As the system learns, when instances are assigned to a particular cluster, the prior probability of that cluster increases which further allows even more instances to be assigned to it. The prior continually increases as all instances eventually are assigned to this cluster. Inversely when, for a moment, no instances are assigned to a particular cluster, a similar vicious cycle emerges and the prior goes down until no instances are ever assigned to it.

Instability can manifest itself in other ways and techniques to predict it for clustering systems such as ours do not exist. We try to avoid it by crafting distance functions and their smoothing carefully and correcting when we notice instability happening.

Another problem arises when clusters are added incrementally and these clusters are used as attributes to instances. The number of attributes in the system is always growing and we are faced with ever growing complexity.

The proximity function is defined probabilistically using Bayes rule with a prior of one to avoid instability due to feedback. Also while most learning systems use the naïve Bayes assumption and assume probabilistic independence, we instead assumed high dependency between cluster features. Since dependency is what creates clusters, assuming independence greatly overestimates instance's cluster membership probabilities. The proximity function is $\Pr(x) = \Pr(x|w)$.

$$\Pr(x|w) = \frac{\Pr(w|x)\Pr(x)}{\Pr(w)}$$

$$\Pr(x) = 1$$

$$\Pr(w|x) / \Pr(w) \approx \text{avg}_{i=1 \rightarrow N} \left(\frac{\Pr(b_i|x)}{\Pr(b_i)} \right)$$

Where x is a cluster from the set of existing clusters $x \in X$. w is the word instance having $\{b_1, \dots, b_N\}$ features.

When an instance's proximity to a cluster is above a certain threshold, the instance is assigned to it.

4 Comparison with previous work

Evaluation is performed by comparing the percentage of dependencies shared between the generated parsed sentences and those of the Suzanne corpus.

Klein and Manning (2004) created a comparable system based on the EM-algorithm which performed with 64.5% accuracy (77.6 F-measure). Our system differs from the Klein and Manning system in that we make use of morphological analysis, stemming and incrementality.

As our research is ongoing, we only have preliminary results for our system. The current configuration achieves accuracy of 27%. This is lower than we expected but there is potential for further experimentation. We expect that by tweaking the clustering step, much greater accuracy can be achieved.

Although our system is based on Yuret (1998), the results are not directly comparable with his work as we chose to use the measure of Klein and Manning. Yuret uses a weak performance measure that only takes into account content words.

5 Future Work

Extensive work is planned for this project which includes trying different proximity functions, changing the threshold where new clusters are created, using a hierarchical clustering scheme instead of flat clusters and better integrating link direction to help the learning.

6 Discussion

The sum of the components of this system interact together to build clusters that represent the information necessary to do dependency parsing. Most importantly, emerging from this clustering structure is a series of nodes that represent learned hidden concepts. The choice of a system that generates such a structure is not a coincidence. There is hope that the clusters form lexical and syntactic categories useful for other tasks.

One important benefit of this system over other systems is that it naturally deals with unseen words without having to use special mechanism.

We hope in the future to be able to exploit the information generated in the clusters to apply the learned grammar to tasks such as semantic disambiguation and machine translation.

References

- Rens Bod, Remko Scha, and Khalil Sima'an, eds. (2003a) *Data-Oriented Parsing*. CSLI Publications, University of Chicago Press. Chicago, IL.
- Rens Bod, Jennifer Hay, Stephanie Jannedy, eds. (2003b) *Probabilistic Linguistics*. MIT Press, Cambridge.
- Michael Collins (1999) *Head-Driven Statistical Models for Natural Language Parsing*. PhD Dissertation, University of Pennsylvania.
- Dan Klein and Christopher D. Manning (2004) Corpus-Based Induction of Syntactic Structure: Models of Dependency and Constituency. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*.
- Roger Levy and Christopher D. Manning (2004) Deep dependencies from context-free statistical parsers: correcting the surface dependency approximation. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*.
- Christopher D. Manning (2003) Probabilistic Syntax. In Bod, R. Hay, J. Jannedy, S. (Ed.) *Probabilistic Linguistics* (pp. 289-343). MIT Press, Cambridge, MA.
- Christopher D. Manning and Hinrich Schütze (1999) *Foundations of Statistical Natural Language Processing*, MIT Press. Cambridge, MA.
- Deniz Yuret (1998) *Discovery of Linguistic Relations Using Lexical Attraction*. PhD Dissertation, Massachusetts Institute of Technology.