

# Overview of Research Work

Cornelia (Karin) Verspoor

## Introduction

During the past four years, I have worked in industry, at start-ups focusing on artificial intelligence and natural language processing products. I have been involved with a wide range of projects, some focused on implementation of solutions to practical NLP issues (for instance, adequate tokenization and end-of-sentence identification for real-world texts, spell-checking, and robust morphological processing), and others with a more research orientation. I have not had the opportunity to publish this work, due to Intellectual Property concerns. However, in this document I will outline the more interesting of these projects, in order to give a flavor of the kind of research I have done. I present each project with a brief abstract.

## Contents

Syntactico-Semantic Clustering of Words .....	2
Grammar rule induction.....	4
Word Sense Discrimination .....	5
Semantic Parsing .....	7
Shallow Parsing .....	9
Experiential Interactive Learning (EIL).....	11
Word Sense Disambiguation.....	14

## Syntactico-Semantic Clustering of Words

(with Ben Goertzel and Onar Åm at Webmind, Inc.)

Syntactic categories are the cornerstone of the representation of syntactic relations used in parsing and semantic analysis – they provide a level of abstraction over groups of words that enables grammatical relations to be efficiently captured. However, this dependency on syntactic categories in grammatical representation introduces the problem of determining valid parts of speech for most words of the lexicon. Current techniques for parsing, whether rule-based or statistical, depend on reliable part-of-speech information to reduce the amount of ambiguity that they must deal with. Simply relying on syntactic constraints introduced through function words is not adequate to achieve reasonable parsing accuracy, other words in the sentence must also be constrained to a small number of syntactic categories.

Given availability of tagged corpora, it is possible to acquire part of speech information for the words represented in those corpora. However, due to the manual effort that is required in creating these corpora, they are often fairly limited in size, and, particularly if bootstrapping techniques are used, may contain tagging errors. Furthermore, each corpus is tagged according to a particular tagset and with certain tagging conventions which reflect a linguist's judgement of appropriate lexical classifications, rather than classifications defined by the grammar itself. To address these problems, we investigate a technique for automatic induction of grammatical categories. The basic assumption underlying the approach is that an understanding of language structure can be inferred through examination of language use.

The strategy which we have implemented involves processing large amounts of text, and collecting data on the neighboring tokens of each token in the text (including punctuation and other non-whitespace characters). The definition of a syntactic category in this context is a set of words which tend to appear in the same positions in sentences. That is, a set of words which tend to appear before or after similar sets of neighboring words. These can be identified through observation of where words appear relative to each other; their patterns of distribution.

In particular, we build *markov chains* for each token, representing the predecessors and successors of that token, and the predecessors and successors of each of those tokens, out to a given neighborhood size (or markov order). An increase in neighborhood size causes an exponential growth in the amount of data that needs to be collected, due to the combinatorics involved, but also enables more fine-grained analysis of contextual similarities between tokens. We chose a relatively small neighborhood size of 3-4 (trigrams or four-grams), following from the notion that most syntactic constraints are relatively localized. We keep count of the number of occurrences of particular chains, in order to calculate their probability relative to the central token.

Once the markov chains have been built, we apply a clustering algorithm to these probabilistic chains in order group together those chains which are most similar to one another (using the sum of differences between two markov chains as the measure of similarity). In this case, we implemented a k-means algorithm which enables specification of a target number of categories (k). Through experimentation we can arrive at a value for k which seems to support interesting and clear categorization of tokens, that is, an appropriate level of syntactic generalization.

The results of this work suggest that this strategy for unsupervised syntactic category induction has promise, in that the clusters arrived at for relatively small values of  $k$  did seem to make linguistic sense. Furthermore, the categories, and the similarity measurements, interestingly reflect some level of semantic categorization in addition to syntactic categorization. For instance, motion verbs tend to be judged as more similar to each other than to other kinds of verbs, due to the similarity in the kinds of words which can appear as subjects for these verbs. We learned several things from this experimentation, however, which suggest alternative strategies that could lead to more useful results:

- The  $k$ -means algorithm is not the most appropriate clustering algorithm for this problem as it does not support soft clustering. This is problematic given that many words have multiple syntactic usages, which are conflated in their markov profiles. Another clustering technique which supports soft clustering, such as the EM algorithm, may lead to better clustering. Perhaps, with soft clustering, it would be possible to take better advantage of the ability of this approach to group words along semantic lines, for instance to distinguish motion verbs from other verbs as suggested previously, or to categorize more effectively for larger values of  $k$ , to support automatic identification of more refined syntactic categories (e.g. 1<sup>st</sup>-person-singular verbs, plural vs. singular nouns, mass vs. count nouns, etc.). With hard clustering, only rough categorizations seem to be effective.
- The kinds of corpora that are available for large-scale processing mainly consist of texts which are linguistically sophisticated, and which therefore contain many syntactically complex structures. The markov profiles for word tokens which are extracted from these texts are often quite "messy" due to the existence of syntactic processes like topicalization, relative clause modification, question inversion and passivization. This makes the interpretation of markov similarity complex, introducing error into the data used for clustering, and therefore into the resulting clusters. Upon reflection, it is clear that an important flaw in using such sophisticated texts for unsupervised learning is that they do not allow the process by which language is learned by human beings to be mimicked in the automatic language learning process. An alternative strategy, more likely to be successful, would be to employ a bootstrapping procedure: first attempting to induce syntactic categories from simple texts, and then using those categories as seeds for the processing of increasingly more complex texts. In conjunction with strategies for hierarchical clustering (rather than the non-hierarchical clustering  $k$ -means provides), a sophisticated category system could be inferred incrementally.

## **Grammar rule induction**

(with Ben Goertzel and Onar Åm at Webmind, Inc.)

Using a strategy based on the same principles as that used for syntactic category discrimination, we made some preliminary investigations into the use of unsupervised methods for grammatical rule inference. The goal of this work was to recognize syntactic category sequences which seem to have similar syntactic distributions, and to characterize those sequences, and in turn combinations of those sequences, formally in terms of grammar rules. These rules should then be able to be applied in order to derive parses of input sentences, indicating the phrasal structure of those sentences. The work builds on ideas in Yuret (1998) for lexically-based models of linguistic relation inference, generalizing his strategy from the word level to the level of syntactic categories, in order to compensate for some of the data sparseness problems which he faced, and to allow the results to be more easily interpretable in terms of standard grammatical representations.

The approach requires the collection of n-grams (markov profiles) at the syntactic category level across a corpus. The result is, for each syntactic category, data reflecting the syntactic contexts in which that category is used. This data can be used as input to a bootstrapping process for grammar rule induction which uses mutual information as a basis for “chunking” pairs of categories. The mutual information between two categories reflects the attraction of those two categories, relative to the other categories surrounding them.

In the first stage of the bootstrapping process, the mutual information between each sequential pair of categories in each input sentence is calculated. When two categories have strong mutual information, they are grouped together and treated as a unit. This unit forms the basis of a phrase, and the category sequence can be viewed as a sequence of terminals which project to a non-terminal; i.e., a grammar rule. We can then begin to collect data on sequences of terminals and non-terminals, or between non-terminals, and iterate the process of identifying structure using mutual information, and feeding this structure back into the data collection process, until higher-order groupings of these categories in sentences is possible.

We augment this process with mechanisms for merging non-terminals based on similar syntactic distributions (represented by the markov profiles). Just as we were able to group words into syntactic categories due to their similar markov profiles, we can group distinct sequences of syntactic categories together due to their distributional similarity, and represent them in terms of a single shared non-terminal. This is necessary in order to achieve hierarchical representations of linguistic structure, i.e. a grammar.

The preliminary results achieved with these techniques suggest that given a very large corpus of text, it is possible to incrementally acquire data which can be used to effectively identify the main phrases in a sentence, and relations between them. However, further research into how well this data corresponds to standard representations of grammar is necessary.

Yuret, Deniz (1998). Discovery of Linguistic Relations Using Lexical Attraction. PhD Dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science.

## Word Sense Discrimination

(with Ben Goertzel, Onar Åm and André Senna at Webmind, Inc.)

Following on from the previously described research in syntactic category discrimination, we investigated the use of corpus-based methods for unsupervised identification of word senses. In this work, rather than focusing on grouping words with similar markov profiles together, we look at how to split the markov profile into groups of markov chains which are similar to one another – effectively clustering individual usages of a given word.

The strategy implemented was based on a simple idea building on bigrams and trigrams:

If PW and WS are good phrases for a given word W, but PWS is not, then W is used in orthogonal senses in the two phrases.

This idea should be effective mainly for distinguishing two senses of a word which fall into distinct syntactic categories, as it is dependent on the grammaticality of particular trigram sequences, which determines the occurrence of those trigrams in the corpus. This observation allows us to extend the basic idea to incorporate measures of similarity among words. In particular, the markov profile similarity which was the foundation of our syntactic category discrimination can be used to provide further evidence for the grammaticality of a particular sequence. That is, if PWS has not been observed in the corpus, but the phrase XWY has been, where P is similar to X and S is similar to Y, this provides evidence for the potential grammaticality of PWS.

We calculate “adherence” between pairs of words, reflecting the probability that the two words occur on either side of the same sense of a third word W. So a predecessor P and a successor S have a certain “adherence” based on whether the phrase PWS occurs more likely than predicted by chance, and its similarity to other phrases which are more likely than predicted by chance. We then cluster the trigrams centered around W according to their similarity with respect to adherence – that is, phrases containing predecessors which adhere to a similar set of successors of W, and successors which adhere to a similar set of predecessors of W. The hope is that these clusters will reflect distinct senses of the word W.

This approach contrasts with work such as Schütze (1998) which uses a much broader notion of context to drive word sense disambiguation, drawing on co-occurrence in a window of size k, where k is significantly larger than the 1-word window we are using for the trigrams here. Whereas those approaches are well-suited for semantically-based sense discrimination, they do not work well for topic-independent senses (i.e. senses which are poor discriminators of particular topics but are rather distributed across a wide range of contexts). To the extent that the topic-independent senses of a word can be distinguished along syntactic lines, our approach should help to fill in the gaps in the semantically-based sense discrimination algorithms.

Using this approach, we were unable to get good word sense discrimination for several syntactically ambiguous words (“train”, “wet”, “tape”, “suit”) tested. The clusters identified show that the algorithm is too dependent on the immediate surface-level variations in trigrams; too many clusters were identified, and the clusters reflected a sensitivity to the occurrence of a particular word as predecessor or successor to the ambiguous word.

Given the primary goal of syntactically-based sense discrimination, a strategy which builds off syntactic analysis more directly would likely lead to more promising results. For instance, given the framework previously outlined for syntactic category induction, the system would have existing models of how words in various syntactic categories behave. These could be used to drive the classification of individual occurrences of a word in a sentence as an occurrence of a particular syntactic category (either through parsing or through more statistically-driven markov profile matching). This would help to overcome the sparse data problem faced by the algorithm suggested here (due to the sparseness of trigrams) by allowing markov data accumulated over a large number of nouns and a large number of verbs to influence the classification of new occurrences. We would thus be able to group instances of ambiguous words according to their syntactic category, giving a syntactically based sense discrimination.

To achieve semantic sense discrimination, a wider context such as that used by Schütze (1998) would be needed. However, given the lack of attention to immediate syntactic context of the semantically-based algorithms, it appears that to handle the full range of word sense phenomena, automatic sense discrimination algorithms should take a hybrid approach, considering both the syntactic and semantic contexts. Such a hybrid approach could actually be viewed as a uniform approach which just differs as to what linguistic level is considered during similarity assessment; where the syntactic context stems from local markov profiles (immediate context windows) and the semantic context derives from broader co-occurrence profiles (wide context windows).

Schütze, H. (1998) Automatic Word Sense Discrimination. *Computational Linguistics*, 24:1.

## Semantic Parsing

(with William Schuler and Michael Ross at Webmind, Inc.)

One of the primary goals of the natural language processing module at Webmind, Inc. was to produce semantic representations reflecting the meaning conveyed by individual sentences in a document. We chose to follow a strategy which incorporated semantic processing into the syntactic parsing stage, where parsing is driven by the requirements of lexical heads, and where the features associated with those lexical heads dictates the semantic structure built during parsing.

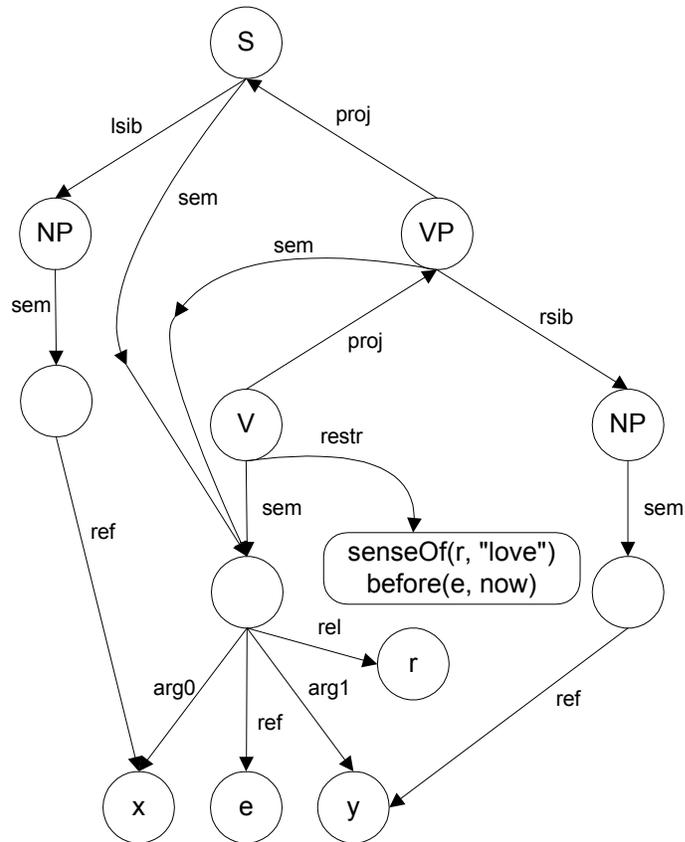
Any head-driven approach to parsing which bundles syntactic and semantic features (such as HPSG) would be suitable for this processing. We have chosen to work with the XTAG framework developed at U. Penn [Joshi, 1985] due to the availability of a large lexicon for English. We extend the syntactic lexicon for XTAG by adding semantic features which form the basis of our semantic processing.

In the XTAG framework, the lexicon consists of feature structures representing extended projections for lexical heads, which essentially define the basic syntactic structure that the word must appear in. These are augmented with transformation rules which define modifications of the basic projections, to support alternative surface level structures such as inversions for queries or passivization, or relative clause modifiers. Parsing proceeds from the bottom-up; phrases are recognized through unification of their projections. The projections can contain specifications for the right or left siblings of a lexical head, which are (attempted to be) unified in the parsing algorithm with the feature structure of an adjacent sibling in the parse tree. Adjunction is handled by propagating the extended projection of the modified node up to the higher phrase structure node that results from the adjunction (this is specified in the lexical feature structure for the adjunctive head).

The integration of syntactic and semantic features into a single feature structure means that the semantics of a phrase is built automatically as a result of unification, through coindexing of appropriate components of the feature structures. As can be seen in Figure 1 below, the feature structure associated with the verb form “loved” specifies the full syntactic context of that verb form, requiring one noun phrase to the left and another to the right, and also dictates the final basic relation structure associated with that verb, indicating that the *referent* of the subject noun phrase is co-indexed with *arg0* of the *relation* expressed by the verb, and the object noun phrase is co-indexed with the relation’s *arg1*. Through unification, those *referents* will be filled in with values indicating specific entities that are involved in the “loved” event.

In addition, it is possible to use the framework to associate semantic constraints with particular syntactic constructions, by adding restrictions as features on the lexical feature structures. For instance, selectional preferences could be represented in terms of restrictions on the arguments of a verb (e.g. specifying that the 1<sup>st</sup> argument must be animate, etc.). This opens the door to allowing pragmatics to interact with syntactic processing, and a parsing process which draws on semantic background knowledge to rule out possible analyses by verifying that certain semantic restrictions hold during parsing. In Figure 1, we see a restriction which draws on the tense information of the verb: the event *e* introduced by this verb is located in the past [ *before(e, now)* ].

Using this “semantic parsing” strategy, we were able to build basic representations of the semantics conveyed by the sentences in our test corpus, handling quite complex syntactic structures with the help of the XTAG-based parsing framework.



**Figure 1:** Lexical entry for the past tense transitive verb "loved."  
 The syntactic information has been simplified for readability.  
 Lower-case labels are used purely for readability.

Joshi, A. K. (1985). How much context sensitivity is necessary for characterizing structural descriptions: Tree adjoining grammars. In *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives*. D. Dowty and A. Zwicky, eds. Cambridge University Press, Cambridge, UK, pp. 206-250.

## Shallow Parsing

One of the largest stumbling blocks in the development of NLP systems capable of some level of semantic interpretation is dependency on syntax. State of the art parsers which rely on large sets of grammar rules and apply those rules probabilistically to generate the most likely parse often have difficulty with ungrammatical inputs, and often cannot reliably produce a single parse (or even several competing parses) for a sentence. In the worst cases, huge amounts of world knowledge and an ability to reason on that world knowledge would be needed in order to resolve certain structural problems such as prepositional phrase attachments.

However, for many practical applications of NLP full parsing of a sentence is not necessary. For text understanding systems (as in the MUC competitions), for instance, it suffices to pick out main events, and the participants in those main events. Modifiers such as adverbs, prepositional phrases and relative clauses are often secondary – they serve to situate particular entities and events in space and time but are not needed to get the “gist” of the event.

For our work at Webmind on document categorization, we were interested in extracting basic events, so that the inputs to our machine learning classification models were much richer semantically – not just looking at the frequencies of particular words in the documents of a category, but also considering relations between the words, at a minimum looking at the frequencies of phrases rather than words (because “New York Times” is an important multi-word term which when broken down into “New”, “York” and “Times” as individual tokens loses its meaning and its import in the document), and eventually considering larger events as a whole (The distinction between “Company X acquires Company Y” and “Company Y acquires Company X” may be relevant for some subtle classifications, for instance). But for this task full syntactic parsing is overkill, as it would generate far more syntactic distinctions than the models would be capable of using.

Similarly, the work on word sense disambiguation at Applied Semantics requires reliable noun- and verb-group identification, so that disambiguation of multi-word terms as well as individual words can be attempted, and so that disambiguation can be biased in favor of senses which correspond to the appropriate part of speech (as in WordNet, concepts are divided into noun, verb, adjective, and adverb groups). So reliable identification of high-level syntactic categories is needed, and basic phrasal structures must be reliably picked out, but fine-grained analysis of syntactic categories or prepositional phrase attachment or even verb argument structure is largely unnecessary.

For both of these applications, then, what we needed was a shallow parser which could robustly handle ungrammatical (real-world) texts to identify basic noun and verb groups. At Webmind, we approached this problem by building off a statistical tagger, and defining a set of regular expressions over sequences of syntactic categories, which were applied in order and were responsible for handling increasingly more complicated structures (in a similar vein to the cascading finite state automata of the SRI FASTUS system [Hobbs et al., 1993]). This particular strategy is limited in terms of how sophisticated the structures picked out can be, since there is no backtracking or undoing of previous decisions possible. It also suffers from errors made in the initial tagging which persist into the phrase identification stage. However, it is fairly effective in the

context of base noun phrases and verb groups which have a fairly predictable structure in English.

At Applied Semantics, we have chosen to incorporate tagging into the shallow parsing step as the results on tagging of the broad syntactic categories we are most interested in are more reliable in a module which takes more context into consideration (since the statistical taggers by and large only use bigrams during category sequence probability maximization). The approach we have taken is to define two finite state automata – one for the noun groups, and one for the verb groups – with transitions possible from one to the other. The module uses the part of speech possibilities for each word in the sentence to guide the transitions from one state to the next.

For instance, if we have a word following a determiner whose part of speech is initially assigned (through *a priori* POS probabilities) to be a verb, we can check to see whether it has an alternative possibility of being a noun, and if so we change its currently assigned part of speech to noun, move to the noun state, and continue with the next word. There are also cases in which a particular part of speech is enforced due to the syntactic context, and where backtracking or, more radically, making an empty transition to the start state to begin parsing a new phrase, is performed depending on the state the system is in and the POS possibilities for the words in the immediate context.

This shallow parsing module is currently under development; however we have already seen tagging results on our test corpus which are somewhat better than those from our previous tagger (based on Brill's transformation-based tagger) for the parts of speech we are interested in.

Jerry R. Hobbs, Douglas Appelt, John Bear, David Israel, Megumi Kameyama, Mark Stickel, and Mabry Tyson. (1993) FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. in Proceedings IJCAI '93, Chambery, France.

## **Experiential Interactive Learning (EIL)**

(with Ben Goertzel, Cate Hartley and Michael Ross at Webmind, Inc.)

At Webmind, Inc. we began work on the design and implementation of an ambitious artificial intelligence project with the goal of defining a framework for learning for a “digital mind” (an AI system, in this case called Webmind) through experience and interaction with other intelligent beings in an environment of mutual meaning. We created a user interface in a limited domain (the File World, a collection of files that Webmind and its human users can mutually manipulate) in which Webmind can get feedback on its actions, can chat with users, and obtain experientially and socially valid groundings for its internal concepts. The goal of the work with respect to language processing is to provide strategies for language learning which are based on the pragmatics of interacting in a shared environment, to enable the system to learn through that interaction, rather than being given explicit sets of external rules.

With respect to language in particular, the foundations of the EIL approach are two simple philosophical principles:

### 1) Language is Mind

Language is not an interface to the mind, it is an aspect of the mind. Learning language is learning linguistic thinking. The more linguistic thinking you've learned, the better you can learn language, and vice versa.

### 2) Language is Social Interaction

Language is fundamentally a way of interacting with other minds. To understand language, a system must understand language as a way of interacting with other minds. To understand language as a way of interacting with other minds, the system must learn language in the context of interacting with other minds.

This approach to language learning is possible only in the context of a system which is equipped with the basic tools for learning and cognition: memory (short- and long-term), perception, action, schema learning (pattern recognition and abstraction), and it was the development of the architecture supporting such cognition which was the primary focus of research at Webmind, Inc.

The strategy which we follow is that learning is guided by a basic motivational structure. Webmind wants to achieve its goals, and its primary goal is to be happy. Webmind therefore must have a way to measure happiness, which initially would be quite simplistic and will depend on feedback from social interaction: Webmind is happy when the entities with which it interacts are happy. This is measured through punishment and reward; users interacting with Webmind provide feedback at every stage of interaction.

The process through which language learning will occur in this framework in many ways works in the opposite direction from standard natural language processing techniques; generally in NLP, we start off with morphology, syntactic tagging, and only move on to semantics when we have reliable syntactic parsing, so that we can use the structural information to drive semantic processing. In EIL, the focus is on semantics, and using language to convey meaning. This mimics human acquisition of language more closely. The first steps in language learning involve learning individual words for particular entities in the shared environment. Then combinations of words will be used to express

more complicated concepts. The rules governing those combinations will be learned through observation and interaction with other “speaking” entities. This is the basis for simple grammar acquisition, which initially will occur on the lexical level, but after processes of reasoning and abstraction will lead to the emergence of syntactic categories and more generalized rules.

This language learning proceeds via the motivational structure: Given the current conversational/viewing-panel/emotional history, Webmind wants to say something that it judges will maximize its happiness. By experimentation and inference, it must discover how to do this. It must study its actions and others’ reactions to its actions, and in this way arrive at its own rules that tell it what to say in what situations.

At the most general level, schema rule learning in an EIL context may be said to recognize three particular types of patterns, which drive the acquisition of language:

- 1) patterns recognized in the perceptual environment, not involving speech acts at all

*Example: when I'm happy, the user becomes happy soon thereafter*

- 2) patterns involving one speech act and other things in the perceptual environment

*Example: the word "data" is often uttered by users when files named \*.dat are present in the environment*

- 3) patterns involving the system's happiness and the system's utterances and users' utterances and/or the perceptual environment

*Example: When the user says "Hi", it makes me happy when I say "Hi" back  
(Note: this is because the user is happy about this, and Webmind is wired to be happy if the user reports happiness)*

*When the user asks a query with "\*.dat" in it and "big" in it, it makes me happy when I respond with the file-size parameter from the file with the name "\*.dat"*

*It makes me happy to utter "bored" when the user hasn't said anything for a while  
(Note: this is because saying "bored" causes the user to speak and user interaction leads to happiness)*

Given the complexity of the language learning task in even a simple shared environment, due to the enormity of the number of different contexts, perceptions, and actions which must be tracked and reasoned on, we developed a strategy for hybridizing the EIL language learning approach with more traditional NLP techniques, which we hoped would jumpstart the language learning process in EIL. It may also have a psychological basis, in that it has been argued that humans are born with some evolutionary linguistic endowment. If that is the case, then it makes perfect sense to provide Webmind with some basic linguistic understanding. In particular, the following types of linguistic knowledge seem appropriate:

- A basic semantic frame for situational understanding: the notion of an agent and a process and a modifier, for example
- The notion of phrasal and clausal structure

- The basic notion of a grammatical rule

We would give the system some basic syntactic categories, some default grammatical rules (without specifying which words fall into those categories, or those structural patterns), and some idea of how syntactic relations correspond to semantic relations (mapping rules), which would give the system a more structured starting point for language learning.

The approach outlined here suggests a methodology for approaching language learning which intermingles language learning with language processing, and with general-purpose cognitive processing. The experiential learning approach is not a strategy for rapid NLP development, but rather an investigation into the mechanisms driving language acquisition in humans, in order to build a language processing system which seamlessly integrates syntax and semantics, in which language use is driven largely by pragmatic conversational and system-internal goals.

## Word Sense Disambiguation

(with Adam Weissman and Gil Elbaz at Applied Semantics)

Most recently, I have been working on word sense disambiguation at Applied Semantics, and algorithms which draw on the results of the disambiguation to enable

- automatic identification of metatags for documents,
- document summarization, and
- document categorization.

The strategy that we are implementing builds on a large ontology, defining words, their meanings, and the relations (both hierarchical and associative) between them, in addition to statistical co-occurrence data collected on the co-occurrence of particular concepts. We have implemented a bootstrapping process which uses only the relations manually represented in the ontology in the first step to drive disambiguation, then collects sense co-occurrence statistics based on results from the first step that are considered to be reliable disambiguations, and feeds this co-occurrence information back into the next cycle of disambiguation, and so on.

The algorithm for disambiguation is to look at a window around each ambiguous word, and to allow all meanings of every word in that window to interact with any meanings of the target word that it has a relation to (relations are used to boost the likelihood of a particular meaning of the word). After several cycles of such interaction across all words in the document, the system settles upon the in-context meaning of each ambiguous word. The basis for this algorithm is the notion of textual coherence: the words in a document, and specifically in a small context around any given word, will tend to cohere semantically, so that the meaning of a word can be predicted from the meanings of the words around it. In a document containing the words “coffee”, “drink”, “sofa”, and the word “Java”, it should be clear that “Java” does not refer to the programming language. This is because the “coffee” sense of “Java” is reinforced by the other words in the document.

We are currently working on ways to improve the results of our disambiguation. This includes incorporating reference resolution to assist disambiguation. For instance, in a document about President George Bush, we want to effectively rule out the possibility that the token “Bush” on its own refers to a plant by recognizing the relationship to the full term. This is particularly effective in documents where named entities occur which are not in the lexicon. We are also working on shallow parsing, as described previously, as a mechanism for improving the part of speech tagging of the words in context, which can be used to bias the a priori probabilities of particular meanings of a word.