Using a Hybrid System of Corpus- and Knowledge-Based Techniques to Automate the Induction of a Lexical Sublanguage Grammar

Geert Jan Wilms
Union University
2447 Hwy 45 Bypass Box 1857. Jackson, TN 38305. USA
jwilms@buster.uu.edu

Abstract

Porting a Natural Language Processing (NLP) system to a new domain remains one of the bottlenecks in syntactic parsing, because of the amount of effort required to fix gaps in the lexicon, and to attune the existing grammar to the idiosyncracies of the new sublanguage. This paper shows how the process of fitting a lexicalized grammar to a domain can be automated to a great extent by using a hybrid system that combines traditional knowledge-based techniques with a corpus-based approach.

1. Porting Bottleneck

The traditional grammar knowledgebase is the product of a never-ending attempt by linguists to impose order on something that refuses to be pinned down because it is a living thing. To a great extent, of course, these linguists are able to point to regularities, because language is first of all a practical thing, a means to communicate, and there must be a common base for such transfer to take place. But all rules have exceptions, and often it turns out these exceptions are not isolated or random, so the rule is finetuned. The problem is that what is "grammatical" depends on the unwritten rules of a certain domain. When the core grammar is augmented to accommodate all these idiosyncracies, the danger is not that an ungrammatical sentence might slip through, but that perfectly legitimate input receives an incorrect analysis that is sanctioned by some peripheral grammar rule that doesn't apply to the domain under investigation. The semantic component which gets this false positive may reject it and request a second reading, and the correct parse will most probably come down the pipeline eventually if the grammar is truly broad-coverage, but a semantic module is not always well equipped to detect such errors and may have a difficult time enough trying to resolve attachment problems, anaphoric references, etc., even when presented with the "right" parse.

In systems that use a lexical grammar, i.e., where part of the grammatical "knowledge" is stored outside the non-terminals of the grammar proper, using subcategorization frames associated with terminals (words in the lexicon), the peril likewise is that this resource becomes bloated over time with options exercised only in certain settings or when the word is used in a marginal sense.

Clearly something must be done to separate the wheat from the chaff; the problem is twofold: getting the grammar and lexicon to a certain level of competence was a laborious and time-consuming process, and undoing this (i.e., eliminating unwanted options) is almost as difficult and painful as the constant augmenting in the first place. And secondly, what constitutes wheat and chaff is different for each domain, so this "dieting" must be repeated for every port.

Corpus-based techniques can help automate this filtering, i.e., the source text should be viewed not only as an "obstacle" to be tamed (parsed), but as a resource that is best authority on what is grammatical for the domain.

2. Data-Driven Attuning

Since the early 90s, there has been a surge of interest in corpus-based NLP research; some researchers have tackled the grammar proper, making it a probabilistic system,[1] or doing away with a rule-based system altogether and inducing a customized grammar from scratch using stochastic methods. Despite the shortcomings of knowledge-based systems, it seems wrong to throw away all that has been gained, imperfect as it is. Rather, a hybrid system should be developed where the strengths of both paradigms are combined. A good example of that is a probabilistic Context Free Grammar.

Both Brent (1993) and Manning (1993), who attempt to induce a lexicon of subcategorization features do so by completely discarding all pre-existing knowledge; both systems are stand-alone, without a parsing engine to test or use the "learned" information. Brent in fact takes the "from scratch" to an extreme, and models his system after the way a child learns to understand language. The algorithm of both authors basically involves a pattern matcher that scans the input for a verb, and once an anchor is found, its right context is searched for cues for subcategorization frames. Brent's cues are very primitive, but because he only picks up frames when the indicators are unambiguous, his results are very reliable, albeit sparse (unless a very large training corpus is used). Manning's triggers on the other hand are more sophisticated, but because they are less dependable he must rely on heavy statistical filtering to reduce the "noise." Although Manning's work in inducing features certainly accomplishes the goal of customizing the lexicon to a particular domain, the

porting process is still very much a manual enterprise in that he must write a mini-parser, a finite state machine that includes an NP recognizer "and various other rules to recognize certain cases that appear frequently" (1993, 237).

The dilemma of any pattern matching approach is in essence a bootstrapping problem; if the goal is to induce syntactic information (in the form of lexical features), then paradoxically some heavy syntactic processing power is needed to "parse" the training data to mine for evidence that a particular verb subcategorizes for an object option, while avoiding false triggers (imposter patterns). Manning has built into his finite state device a panic mode to skip over ambiguous elements, but the trick is to recognize when things get hairy; that is where a lot of programming effort takes place, and this finetuning is never over (and must be repeated for every port to a new domain) as Manning himself admits (1993, 238).

3. Category Space of Context Digests

The category space described in this paper uses a very different approach to induce subcategorization frames; instead of starting from scratch, the existing rich lexicon is exploited and features are assigned to new words based on their paradigmatic relatedness to known words. Thus instead of having to "hunt" for evidence, this approach is able to exploit the expertise of seasoned linguists who constructed the initial lexicon, which was intentionally designed to be broad-coverage. Such a strategy not only avoids having to distinguish good cues from irrelevant triggers, but is capable of inducing some features like ASSERTION for which there is no marker that would indicate its presence.

A category space is a multi-dimensional space in which the syntactic category of words is represented by a vector of co-occurrence counts (Schütze 1993). Proximity between two such vectors, or context digests, can be used to measure the paradigmatic relatedness of the words they represent (Schütze and Pedersen 1993). Paradigmatic relatedness indicates how well two words can be substituted for each other, i.e., how similar their syntactic behavior is. This is not the same as the synonym relationship, which is based on semantic similarity.

There are two general approaches in the literature to collecting distributional information: window-based and syntactically-based (Charniak 1993). In the latter scheme the text is scanned until a section is found that is deemed to be relevant. The "rough" structure of the sentence is computed, a process known as partial parsing. This produces a flat tree with phrase boundaries marked and identified by type, but without much internal detail.

A second approach to collecting relevant distributional information is to keep co-occurrence counts of the nearest lexical neighbors of a word, usually within a fixed distance or "window." Markov models, for example, predict the POS of a word based on the tags of the two or three words preceding it (bigrams and trigrams respectively). Schütze has experimented with window lengths of four words, two hundred letter fourgrams and two thousand characters (Schütze 1993).

In the research presented here, a window of four was adopted, i.e., for words of interest in the domain of physical chemistry, co-occurrence counts were kept between those words and their immediate left neighbors ($w_{i-1}w_i$), immediate right neighbors ($w_i\ w_{i+1}$), and left and right neighbors that are two words away ($w_{i-2}\ w_i$ and $w_i\ w_{i+2}$ respectively).

One importance difference between the category space reported here from the one in Schütze and Pedersen (1993) is that words were disambiguated by part of speech so as not to mix up context information of unrelated tokens, a problem Schütze acknowledges plagues his system (1993, 254). The corpus was tagged using Brill's tagger (Brill 1993), which is based on what he calls transformation-based error-driven learning. 1430 word types tagged as verbs occurred frequently enough (>10x) in the training corpus to warrant constructing a vector or context digest. As Zipf's law would predict, there is a long tail of word types which occur too infrequently to permit gathering useful statistics.

Each window of the context digests tracks co-occurrence counts with word types of any POS, provided these types have a minimum frequency of 100 in the training corpus. For "rare" neighbors, the algorithm simply records the neighbor's POS, a compromise to keep the size of the arrays manageable, while providing some information on the syntactic context.

Context digests are formed by combining the 4 fixed windows, each consisting of co-occurrence counts with 5,509 possible neighbors. In addition, some limited long(er)-distance information is appended to the vector: the training corpus has been augmented with bracketing information, that is, with implicit trees that exhibit binary branching, but whose nonterminals are unlabelled. This is another application of Brill's transformation-based error-driven learner (Brill 1993), which was trained on 32,000 bracketed sentences from the Penn Treebank. These phrasal boundaries are of variable length, and can in fact span the whole sentence. Ideally, the name of the type phrase that the verb occurred in should be used as a clustering feature, but since this information is unavailable (the non-terminals in the trees implicit in the bracketing are unlabelled) the next best thing is used, and each boundary is marked by a pair of tags occurring on either side of the bracket.

Each context digest for verbs, then, contains

27,654 possible entries. The resulting matrix is very sparse, however; the density for the verb category space is only 1.5 percent. Hence the distributional information is generalized by means of a matrix manipulation method called Singular Value Decomposition (SVD). This technique is often used in factor analysis, because reducing the representation to a low dimensionality allows one to better visualize the space. It is exactly this compactness of representation that has led Schütze to apply SVD to the field of NLP, to reduce the number of input parameters to a neural net, without sacrificing too many of the fine distinctions in the original text (Schütze 1993). Deerweester et al. (1990) introduced SVD to the field of information retrieval for improved document representations; the original term-document matrix is decomposed into linearly independent factors, many of which are very small. An approximate model with fewer dimensions can be constructed by ignoring these small components. By combining only the first k linearly independent components, a reduced model is built which disregards lesser terminology variations, because k is smaller than the number of rows (terms).

To generalize the associational patterns in the category space that was bootstrapped from the physical chemistry corpus, SVD was applied with a conservative value for k of 350. The tool used for this purpose was a slightly modified version of the las2 module from the SVDPACKC package (Berry et al. 1993). The generalizing effect of SVD causes the category space for verbs to become much less sparse: 35.4 percent of the entries now have non-zero "counts." Most of these are new counts, i.e. SVD infers context similarities between words that may not be apparent in the original co-occurrence matrix due to the natural randomness in any corpus sample. The average number of context digests that are very similar (greater than 97 percent confidence) remains fairly constant after SVD, but the dimension reduction provides a lot more information about syntactic behavior when a less strict cutoff value is adopted (say 90 percent).

4. Induction based on Neighborhoods

Proximity in this reduced space is then used to find for all the context digests a neighborhood of words that are paradigmatically related. Proximity can be computed by using the cosine similarity measure, which was a major feature of the SMART information retrieval system (Salton 1983). This measures the cosine of the angle between two context digests, which can be viewed as vectors in a s-dimensional space.

The category space can be clustered by comparing pairs of context digests using the cosine similarity measure; such clusters contain words whose syntactic behavior is substantially similar. The degree of similarity depends on the adopted threshold value.

However, these neighborhoods are not traditional clusters; each verb has its own individual representation in a multi-dimensional space, i.e. is the center of its own neighborhood. Typically any given verb is a vector which simultaneously belongs in several neighborhoods.

Verbal subcategorization frames like transitivity, or the ability to take a that-complement or to-infinitive can be induced for new words based on a "composite" of features associated with "similar" verbs that are defined in the lexicon. The knowledgebase used in this research is the domain-independent lexicon of PUNDIT, a broad-coverage symbolic NLP system, which contains 164 verbs with detailed subcategorization information (Hirschman et al. 1989). PUNDIT's features are a subset of Sager's Linguistic String Project (Sager 1981), which include selectional restrictions, features that license constructs, and object options that affect the interpretation of a sentence.

The induction works as follows: each verb has its own neighborhood, formed by computing the cosine similarity weight between it and all other verbs in the category space, and by retaining those whose weight exceeds a certain threshold. If there are no nearby verbs with known features, more remote words can be used for deciding on whether a certain feature should apply to the verb being examined, especially if a substantial majority of these "distant relatives" are in agreement. If the features are treated as boolean values (present/not present), it will most certainly happen in neighborhoods with liberal cutoff points that there will be some disagreement for individual options, so a heuristic must negotiate these "conflicts" and settle for the best abstraction. Such a heuristic should have the following three characteristics:

1) verbs that are close to the word being examined should carry more weight in the decision process than verbs that are closer to the perimeter.
2) both positive and negative evidence (the absence of a feature for a particular verb) should be considered.
3) given the fact that the presence of a feature is the result of a positive decision/action (by a linguist), whereas the absence may be an oversight, there should be a (slight) bias in favor of the former; the sensitivity threshold can be adjusted by shifting the point at which the weight of evidence is considered sufficient to decide in favor of adopting the feature.

The existing verbs in the lexicon themselves undergo a similar process whereby they are fitted to the domain: some of their "generic" features which are not appropriate are dropped, whereas "gaps" in object options are filled. The net result is that the grammar

becomes attuned to the sublanguage: parses become possible because the enabling features are present, while the search space is pruned of many false positives because unnecessary features are omitted.

5. Evaluation

Manning evaluates his system by computing precision and recall scores with the OALD dictionary as golden standard. However, precision is not a good yardstick for evaluating the performance of the induction process, because it measures the outcome against a "flawed" lexicon; the induced features, because of the data-driven nature of the process, are more "precise" when measured against the "real world" of the sublanguage domain than the hand-built entries that are the product mostly of introspection and anecdotal evidence. The system described in this paper was tested instead by comparing the number of successful parses of a held-out test corpus before and after customizing the lexicon. Out-of-the-box PUNDIT returned 42 parses for the 170 sentences in the training corpus (some of which were false positives), versus 94 successful parses using the attuned lexicon. It should be pointed out that these 94 sentences contain an average of 2.14 verbs.

6. Conclusion

The category space is the arbiter of paradigmatic relatedness, and since it is bootstrapped from a training corpus that is representative for the domain sublanguage, the resulting lexical entries will be customized for that domain. Porting the lexicon to a new domain is as simple as bootstrapping another category space. Experiments with PUNDIT, a broad-coverage symbolic NLP system, have shown that the category space can successfully be used to induce features like transitivity and subcategorization for clauses and infinitival complements.

The advantage of combining data-driven mining with the existing lexical knowledgebase over other bootstrapping methods is that this approach does not require the manual identification of appropriate cues for subcategorization features, or the involved construction of a pattern matcher that is sophisticated enough to ignore false triggers.

7. References

Berry, Michael, Theresa Do, Gavin O'Brien, Vijay Krishna, and Sowmini Varadhan. 1993. SVDPACKC (version 1.0) user's guide. Knoxville, TN: Department of Computer Science, University of Tennessee. Technical Report, CS-93-194.

Brent, Michael. 1993. From grammar to lexicon. Unsupervised learning of lexical syntax. Computational Linguistics: Special Report on Using Large Corpora : II 19 (June): 243-62.

Brill, Eric. 1993. A corpus-based approach to language learning. Ph.D. diss., University of Pennsylvania.

Charniak, Eugene. 1993. Statistical language learning. Cambridge, MA: MIT Press.

Deerweester, Scott, Susan Dumais, George Furnas, Thomas Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. Journal of the American Society for Information Science 41 (September): 391-407.

Hirschman, Lynette, Martha Palmer, John Dowding, Deborah Dahl, Marcia Linebarger, Rebecca Passonneau, François Lang, Catherine Ball, and Carl Weir. 1989. The PUNDIT natural language processing system. In Proceedings of the annual artificial intelligence systems in government conference held in Washington, D.C., March, 1989.

Manning, Christopher. 1993. Automatic acquisition of a large subcategorization dictionary from corpora. In Proceedings of the thirty-first annual meeting of the ACL held in Columbus, OH, June, 1993, by the Association for Computational Linguistics. 235-42.

Sager, Naomi, ed. 1981. Natural language information processing: A computer grammar of English and its applications. Reading, MA: Addison-Wesley.

Salton, Gerald. 1983. Introduction to modern information retrieval. New York, N.Y.: McGraw-Hill.

Schütze, Hinrich. 1993. Part-of-speech induction from scratch. In Proceedings of the thirty-first annual meeting of the ACL held in Columbus, OH, June, 1993, by the Association for Computational Linguistics. 251-58.

Schütze, Hinrich, and Jan Pedersen. 1993. A vector model for syntagmatic and paradigmatic relatedness. In Proceedings of the ninth annual conference of the Centre of the new OED and Text Research held in Oxford, England, 1993, by the University of Waterloo.

Wilms, G. Jan. 1995. Automated induction of a lexical sublanguage grammar using a hybrid system of corpus- and knowledge-based techniques. Ph.D. diss., Mississippi State University.