Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction

Brian Roark
Cognitive and Linguistic Sciences
Box 1978
Brown University
Providence, RI 02912, USA
Brian_Roark@Brown.edu

Eugene Charniak
Computer Science
Box 1910
Brown University
Providence, RI 02912, USA
ec@cs.brown.edu

Abstract

Generating semantic lexicons semi-automatically could be a great time saver, relative to creating them by hand. In this paper, we present an algorithm for extracting potential entries for a category from an on-line corpus, based upon a small set of exemplars. Our algorithm finds more correct terms and fewer incorrect ones than previous work in this area. Additionally, the entries that are generated potentially provide broader coverage of the category than would occur to an individual coding them by hand. Our algorithm finds many terms not included within Wordnet (many more than previous algorithms), and could be viewed as an "enhancer" of existing broad-coverage resources.

1 Introduction

Semantic lexicons play an important role in many natural language processing tasks. Effective lexicons must often include many domain-specific terms, so that available broad coverage resources, such as Wordnet (Miller, 1990), are inadequate. For example, both *Escort* and *Chinook* are (among other things) types of vehicles (a car and a helicopter, respectively), but neither are cited as so in Wordnet. Manually building domain-specific lexicons can be a costly, time-consuming affair. Utilizing existing resources, such as on-line corpora, to aid in this task could improve performance both by decreasing the time to construct the lexicon and by improving its quality.

Extracting semantic information from word co-occurrence statistics has been effective, particularly for sense disambiguation (Schütze, 1992; Gale et al., 1992; Yarowsky, 1995). In Riloff and Shepherd (1997), noun co-occurrence statistics were used to indicate nominal category membership, for the purpose of aiding in the construction of semantic lexicons. Generically, their algorithm can be outlined as follows:

1. For a given category, choose a small set of exemplars (or 'seed words')
2. Count co-occurrence of words and seed words within a corpus
3. Use a figure of merit based upon these counts to select new seed words
4. Return to step 2 and iterate *n* times
5. Use a figure of merit to rank words for category membership and output a ranked list

Our algorithm uses roughly this same generic structure, but achieves notably superior results, by changing the specifics of: what counts as co-occurrence; which figures of merit to use for new seed word selection and final ranking; the method of initial seed word selection; and how to manage compound nouns. In sections 2-5 we will cover each of these topics in turn. We will also present some experimental results from two corpora, and discuss criteria for judging the quality of the output.

2 Noun Co-Occurrence

The first question that must be answered in investigating this task is why one would expect it to work at all. Why would one expect that members of the same semantic category would co-occur in discourse? In the word sense disambiguation task, no such claim is made: words can serve their disambiguating purpose regardless of part-of-speech or semantic characteristics. In motivating their investigations, Riloff and Shepherd (henceforth R&S) cited several very specific noun constructions in which co-occurrence between nouns of the same semantic class would be expected, including conjunctions (cars and trucks), lists (planes, trains, and automobiles), appositives (the plane, a twin-engined Cessna) and noun compounds (pickup truck).

Our algorithm focuses exclusively on these constructions. Because the relationship between nouns in a compound is quite different than that between nouns in the other constructions, the algorithm consists of two separate components: one to deal with conjunctions, lists, and appositives; and the other to deal with noun compounds. All compound nouns in the former constructions are represented by the head of the compound. We made the simplifying assumptions that a compound noun is a string of consecutive nouns (or, in certain cases, adjectives - see discussion below), and that the head of the compound is the rightmost noun.

To identify conjunctions, lists, and appositives, we first parsed the corpus, using an efficient statistical parser (Charniak et al., 1998), trained on the Penn Wall Street Journal Treebank (Marcus et al., 1993). We defined co-occurrence in these constructions using the standard definitions of dominance and precedence. The relation is stipulated to be transitive, so that all head nouns in a list co-occur with each other (e.g. in the phrase *planes, trains, and automobiles* all three nouns are counted as co-occuring with each other). Two head nouns co-occur in this algorithm if they meet the following four conditions:

1. they are both dominated by a common NP node
2. no dominating S or VP nodes are dominated by that same NP node
3. all head nouns that precede one, precede the other
4. there is a comma or conjunction that precedes one and not the other

In contrast, R&S counted the closest noun to the left and the closest noun to the right of a head noun as co-occuring with it. Consider the following sentence from the MUC-4 (1992) corpus: *"A cargo aircraft may drop bombs and a truck may be equipped with artillery for war."* In their algorithm, both *cargo* and *bombs* would be counted as co-occuring with *aircraft*. In our algorithm, co-occurrence is only counted within a noun phrase, between head nouns that are separated by a comma or conjunction. If the sentence had read: *"A cargo aircraft, fighter plane, or combat helicopter ..."*, then *aircraft*, *plane*, and *helicopter* would all have counted as co-occuring with each other in our algorithm.

3 Statistics for selecting and ranking

R&S used the same figure of merit both for selecting new seed words and for ranking words in the final output. Their figure of merit was simply the ratio of the times the noun coocurs with a noun in the seed list to the total frequency of the noun in the corpus. This statistic favors low frequency nouns, and thus necessitates the inclusion of a minimum occurrence cutoff. They stipulated that no word occuring fewer than six times in the corpus would be considered by the algorithm. This cutoff has two effects: it reduces the noise associated with the multitude of low frequency words, and it removes from consideration a fairly large number of certainly valid category members. Ideally, one would like to reduce the noise without reducing the number of valid nouns. Our statistics allow for the inclusion of rare occcurances. Note that this is particularly important given our algorithm, since we have restricted the relevant occurrences to a specific type of structure; even relatively common nouns may not occur in the corpus more than a handful of times in such a context.

The two figures of merit that we employ, one to select and one to produce a final rank, use the following two counts for each noun:

1. a noun's co-occurrences with seed words
2. a noun's co-occurrences with any word

To select new seed words, we take the ratio of count 1 to count 2 for the noun in question. This is similar to the figure of merit used in R&S, and also tends to promote low frequency nouns. For the final ranking, we chose the log likelihood statistic outlined in Dunning (1993), which is based upon the co-occurrence counts of all nouns (see Dunning for details). This statistic essentially measures how surprising the given pattern of co-occurrence would be if the distributions were completely random. For instance, suppose that two words occur forty times each, and they co-occur twenty times in a million-word corpus. This would be more surprising for two completely random distributions than if they had each occurred twice and had always co-occurred. A simple probability does not capture this fact.

The rationale for using two different statistics for this task is that each is well suited for its particular role, and not particularly well suited to the other. We have already mentioned that the simple ratio is ill suited to dealing with infrequent occurrences. It is thus a poor candidate for ranking the final output, if that list includes words of as few as one occurrence in the corpus. The log likelihood statistic, we found, is poorly suited to selecting new seed words in an iterative algorithm of this sort, because it promotes high frequency nouns, which can then overly influence selections in future iterations, if they are selected as seed words. We termed this phenomenon *infection*, and found that it can be so strong as to kill the further progress of a category. For example, if we are processing the category *vehicle* and the word *artillery* is selected as a seed word, a whole set of weapons that cooccur with artillery can now be selected in future iterations. If one of those weapons occurs frequently enough, the scores for the words that it co-occurs with may exceed those of any vehicles, and this effect may be strong enough that no vehicles are selected in any future iteration. In addition, because it promotes high frequency terms, such a statistic tends to have the same effect as a minimum occurrence cutoff, i.e. few if any low frequency words get added. A simple probability is a much more conservative statistic, insofar as it selects far fewer words with the potential for infection, it limits the extent of any infection that does occur, and it includes rare words. Our motto in using this statistic for selection is, "First do no harm."

4 Seed word selection

The simple ratio used to select new seed words will tend not to select higher frequency words in the category. The solution to this problem is to make the initial seed word selection from among the most frequent head nouns in the corpus. This is a sensible approach in any case, since it provides the broadest coverage of category occurrences, from which to select additional likely category members. In a task that can suffer from sparse data, this is quite important. We printed a list of the most common nouns in the corpus (the top 200 to 500), and selected category members by scanning through this list. Another option would be to use head nouns identified in Wordnet, which, as a set, should include the most common members of the category in question. In general, however, the strength of an algorithm of this sort is in identifying infrequent or specialized terms. Table 1 shows the seed words that were used for some of the categories tested.

5 Compound Nouns

The relationship between the nouns in a compound noun is very different from that in the other constructions we are considering. The non-head nouns in a compound noun may or may not be legitimate members of the category. For instance, either *pickup truck* or *pickup* is a legitimate vehicle, whereas *cargo plane* is legitimate, but *cargo* is not. For this reason, co-occurrence within noun compounds is not considered in the iterative portions of our algorithm. Instead, all noun compounds with a head that is included in our final ranked list, are evaluated for inclusion in a second list.

The method for evaluating whether or not to include a noun compound in the second list is intended to exclude constructions such as *government plane* and include constructions such as *fighter plane*. Simply put, the former does not correspond to a type of vehicle in the same way that the latter does. We made the simplifying assumption that the higher the probability of the head given the non-head noun, the better the construction for our purposes. For instance, if the noun *government* is found in a noun compound, how likely is the head of that compound to be *plane*? How does this compare to the noun *fighter*?

For this purpose, we take two counts for each noun in the compound:

1. The number of times the noun occurs in a noun compound with each of the nouns to its right in the compound
2. The number of times the noun occurs in a noun compound

For each non-head noun in the compound, we

Crimes (MUC):	murder(s), crime(s), killing(s), trafficking, kidnapping(s)
Crimes (WSJ):	murder(s), crime(s), theft(s), fraud(s), embezzlement
Vehicle:	plane(s), helicopter(s), car(s), bus(es), aircraft(s), airplane(s), vehicle(s)
Weapon:	bomb(s), weapon(s), rifle(s), missile(s), grenade(s), machinegun(s), dynamite
Machines:	computer(s), machine(s), equipment, chip(s), machinery

Table 1: Seed Words Used

evaluate whether or not to omit it in the output. If all of them are omitted, or if the resulting compound has already been output, the entry is skipped. Each noun is evaluated as follows:

First, the head of that noun is determined. To get a sense of what is meant here, consider the following compound: *nuclear-powered aircraft carrier*. In evaluating the word *nuclear-powered*, it is unclear if this word is attached to *aircraft* or to *carrier*. While we know that the head of the entire compound is *carrier*, in order to properly evaluate the word in question, we must determine which of the words following it is its head. This is done, in the spirit of the Dependency Model of Lauer (1995), by selecting the noun to its right in the compound with the highest probability of occuring with the word in question when occurring in a noun compound. (In the case that two nouns have the same probability, the rightmost noun is chosen.) Once the head of the word is determined, the ratio of count 1 (with the head noun chosen) to count 2 is compared to an empirically set cutoff. If it falls below that cutoff, it is omitted. If it does not fall below the cutoff, then it is kept (provided its head noun is not later omitted).

6 Outline of the algorithm

The input to the algorithm is a parsed corpus and a set of initial seed words for the desired category. Nouns are matched with their plurals in the corpus, and a single representation is settled upon for both, e.g. *car(s)*. Co-Occurrence bigrams are collected for head nouns according to the notion of co-occurrence outlined above. The algorithm then proceeds as follows:

1. Each noun is scored with the selecting statistic discussed above.
2. The highest score of all non-seed words is determined, and all nouns with that score are added to the seed word list. Then return to step one and repeat. This iteration continues many times, in our case fifty.
3. After the number of iterations in (2) are completed, any nouns that were not selected as seed words are discarded. The seed word set is then returned to its original members.
4. Each remaining noun is given a score based upon the log likelihood statistic discussed above.
5. The highest score of all non-seed words is determined, and all nouns with that score are added to the seed word list. We then return to step (5) and repeat the same number of times as the iteration in step (2).
6. Two lists are output, one with head nouns, ranked by when they were added to the seed word list in step (6), the other consisting of noun compounds meeting the outlined criterion, ordered by when their heads were added to the list.

7 Empirical Results and Discussion

We ran our algorithm against both the MUC-4 corpus and the Wall Street Journal (WSJ) corpus for a variety of categories, beginning with the categories of *vehicle* and *weapon*, both included in the five categories that R&S investigated in their paper. Other categories that we investigated were *crimes*, *people*, *commercial sites*, *states* (as in static states of affairs), and *machines*. This last category was run because of the sparse data for the category *weapon* in the Wall Street Journal. It represents roughly the same kind of category as weapon, namely technological artifacts. It, in turn, produced sparse results with the MUC-4 corpus. Tables 3 and 4 show the top results on both the head noun and the compound noun lists generated for the categories we tested.

R&S evaluated terms for the degree to which they are related to the category. In contrast, we counted valid only those entries that are clear members of the category. Related words (e.g.

crash for the category *vehicle*) did not count. A valid instance was: (1) novel (i.e. not in the original seed set); (2) unique (i.e. not a spelling variation or pluralization of a previously encountered entry); and (3) a proper class within the category (i.e. not an individual instance or a class based upon an incidental feature). As an illustration of this last condition, neither *Galileo Probe* nor *gray plane* is a valid entry, the former because it denotes an individual and the latter because it is a class of planes based upon an incidental feature (color).

In the interests of generating as many valid entries as possible, we allowed for the inclusion in noun compounds of words tagged as adjectives or cardinality words. In certain occasions (e.g. *four-wheel drive truck* or *nuclear bomb*) this is necessary to avoid losing key parts of the compound. Most common adjectives are dropped in our compound noun analysis, since they occur with a wide variety of heads.

We determined three ways to evaluate the output of the algorithm for usefulness. The first is the ratio of valid entries to total entries produced. R&S reported a ratio of .17 valid to total entries for both the *vehicle* and *weapon* categories (see table 2). On the same corpus, our algorithm yielded a ratio of .329 valid to total entries for the category *vehicle*, and .36 for the category *weapon*. This can be seen in the slope of the graphs in figure 1. Tables 2 and 5 give the relevant data for the categories that we investigated. In general, the ratio of valid to total entries fell between .2 and .4, even in the cases that the output was relatively small.

A second way to evaluate the algorithm is by the total number of valid entries produced. As can be seen from the numbers reported in table 2, our algorithm generated from 2.4 to nearly 3 times as many valid terms for the two contrasting categories from the MUC corpus than the algorithm of R&S. Even more valid terms were generated for appropriate categories using the Wall Street Journal.

Another way to evaluate the algorithm is with the number of valid entries produced that are not in Wordnet. Table 2 presents these numbers for the categories *vehicle* and *weapon*. Whereas the R&S algorithm produced just 11 terms not already present in Wordnet for the two categories combined, our algorithm produced 106, or over 3 for every 5 valid terms produced. It is for this reason that we are billing our algorithm as something that could enhance existing broad-coverage resources with domain-specific lexical information.

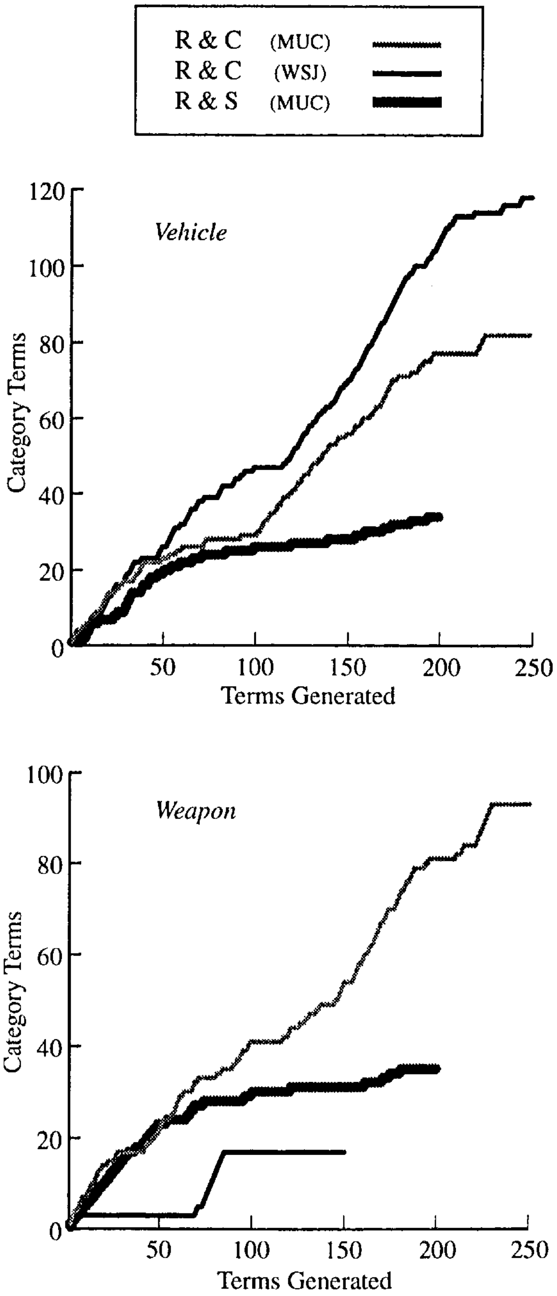

Figure 1: Results for the Categories Vehicle and Weapon

8 Conclusion

We have outlined an algorithm in this paper that, as it stands, could significantly speed up

		MUC-4 corpus			WSJ corpus		
Category	**Algorithm**	**Total Terms Generated**	**Valid Terms Generated**	**Valid Terms not in Wordnet**	**Total Terms Generated**	**Valid Terms Generated**	**Valid Terms not in Wordnet**
Vehicle	R & C	249	82	52	339	123	81
Vehicle	R & S	200	34	4	NA	NA	NA
Weapon	R & C	257	93	54	150	17	12
Weapon	R & S	200	34	7	NA	NA	NA

Table 2: Valid category terms found that are not in Wordnet

Crimes (a): terrorism, extortion, robbery(es), assassination(s), arrest(s), disappearance(s), violation(s), assault(s), battery(es), tortures, raid(s), seizure(s), search(es), persecution(s), siege(s), curfew, capture(s), subversion, good(s), humiliation, evictions, addiction, demonstration(s), outrage(s), parade(s)
Crimes (b): action-the murder(s), Justines crime(s), drug trafficking, body search(es), dictator Noriega, gun running, witness account(s)
Sites (a): office(s), enterprise(s), company(es), dealership(s), drugstore(s), pharmacies, supermarket(s), terminal(s), aqueduct(s), shoeshops, marinas, theater(s), exchange(s), residence(s), business(es), employment, farmland, range(s), industry(es), commerce, etc., transportation-have, market(s), sea, factory(es)
Sites (b): grocery store(s), hardware store(s), appliance store(s), book store(s), shoe store(s), liquor store(s), Albatros store(s), mortgage bank(s), savings bank(s), creditor bank(s), Deutsch-Suedamerikanische bank(s), reserve bank(s), Democracia building(s), apartment building(s), hospital-the building(s)
Vehicle (a): gunship(s), truck(s), taxi(s), artillery, Hughes-500, tires, jitneys, tens, Huey-500, combat(s), ambulance(s), motorcycle(s), Vides, wagon(s), Huancora, individual(s), KFIR, M-5S, T-33, Mirage(s), carrier(s), passenger(s), luggage, firemen, tank(s)
Vehicle (b): A-37 plane(s), A-37 Dragonfly plane(s), passenger plane(s), Cessna plane(s), twin-engined Cessna plane(s), C-47 plane(s), gray plane(s), KFIR plane(s), Avianca-HK1803 plane(s), LATN plane(s), Aeronica plane(s), 0-2 plane(s), push-and-pull 0-2 plane(s), push-and-pull plane(s), fighter-bomber plane(s)
Weapon (a): launcher(s), submachinegun(s), mortar(s), explosive(s), cartridge(s), pistol(s), ammunition(s), carbine(s), radio(s), amount(s), shotguns, revolver(s), gun(s), materiel, round(s), stick(s), clips, caliber(s), rocket(s), quantity(es), type(s), AK-47, backpacks, plugs, light(s)
Weapon (b): car bomb(s), night-two bomb(s), nuclear bomb(s), homemade bomb(s), incendiary bomb(s), atomic bomb(s), medium-sized bomb(s), highpower bomb(s), cluster bomb(s), WASP cluster bomb(s), truck bomb(s), WASP bomb(s), high-powered bomb(s), 20-kg bomb(s), medium-intensity bomb(s)

Table 3: Top results from (a) the head noun list and (b) the compound noun list using MUC-4 corpus

the task of building a semantic lexicon. We have also examined in detail the reasons why it works, and have shown it to work well for multiple corpora and multiple categories. The algorithm generates many words not included in broad coverage resources, such as Wordnet, and could be thought of as a Wordnet "enhancer" for domain-specific applications.

More generally, the relative success of the algorithm demonstrates the potential benefit of narrowing corpus input to specific kinds of constructions, despite the danger of compounding sparse data problems. To this end, parsing is invaluable.

9 Acknowledgements

Thanks to Mark Johnson for insightful discussion and to Julie Sedivy for helpful comments.

References

E. Charniak, S. Goldwater, and M. Johnson. 1998. Edge-based best-first chart parsing. forthcoming.

T. Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

W.A. Gale, K.W. Church, and D. Yarowsky. 1992. A method for disambiguating word

Crimes (a): conspiracy(es), perjury, abuse(s), influence-peddling, sleaze, waste(s), forgery(es), inefficiency(es), racketeering, obstruction, bribery, sabotage, mail, planner(s), burglary(es), robbery(es), auto(s), purse-snatchings, premise(s), fake, sin(s), extortion, homicide(s), killing(s), statute(s)
Crimes (b): bribery conspiracy(es), substance abuse(s), dual-trading abuse(s), monitoring abuse(s), dessert-menu planner(s), gun robbery(es), chance accident(s), carbon dioxide, sulfur dioxide, boiler-room scam(s), identity scam(s), 19th-century drama(s), fee seizure(s)
Machines (a): workstation(s), tool(s), robot(s), installation(s), dish(es), lathes, grinders, subscription(s), tractor(s), recorder(s), gadget(s), bakeware, RISC, printer(s), fertilizer(s), computing, pesticide(s), feed, set(s), amplifier(s), receiver(s), substance(s), tape(s), DAT, circumstances
Machines (b): hand-held computer(s), Apple computer(s), upstart Apple computer(s), Apple MacIntosh computer(s), mainframe computer(s), Adam computer(s), Cray computer(s), desktop computer(s), portable computer(s), laptop computer(s), MIPS computer(s), notebook computer(s), mainframe-class computer(s), Compaq computer(s), accessible computer(s)
Sites (a): apartment(s), condominium(s), tract(s), drugstore(s), setting(s), supermarket(s), outlet(s), cinema, club(s), sport(s), lobby(es), lounge(s), boutique(s), stand(s), landmark, bodegas, thoroughfare, bowling, steak(s), arcades, food-production, pizzerias, frontier, foreground, mart
Sites (b): department store(s), flagship store(s), warehouse-type store(s), chain store(s), five-and-dime store(s), shoe store(s), furniture store(s), sporting-goods store(s), gift shop(s), barber shop(s), film-processing shop(s), shoe shop(s), butcher shop(s), one-person shop(s), wig shop(s)
Vehicle (a): truck(s), van(s), minivans, launch(es), nightclub(s), troop(s), october, tank(s), missile(s), ship(s), fantasy(es), artillery, fondness, convertible(s), Escort(s), VII, Cherokee, Continental(s), Taurus, jeep(s), Wagoneer, crew(s), pickup(s), Corsica, Beretta
Vehicle (b): gun-carrying plane(s), commuter plane(s), fighter plane(s), DC-10 series-10 plane(s), high-speed plane(s), fuel-efficient plane(s), UH-60A Blackhawk helicopter(s), passenger car(s), Mercedes car(s), American-made car(s), battery-powered car(s), battery-powered racing car(s), medium-sized car(s), side car(s), exciting car(s)

Table 4: Top results from (a) the head noun list and (b) the compound noun list using WSJ corpus

	MUC-4 corpus		WSJ corpus	
Category	**Total Terms**	**Valid Terms**	**Total Terms**	**Valid Terms**
Crimes	115	24	90	24
Machines	0	0	335	117
People	338	85	243	103
Sites	155	33	140	33
States	90	35	96	17

Table 5: Valid category terms found by our algorithm for other categories tested

senses in a large corpus. *Computers and the Humanities*, 26:415–439.

M. Lauer. 1995. Corpus statistics meet the noun compound: Some empirical results. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 47–55.

M.P. Marcus, B. Santorini, and M.A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

G. Miller. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4).

MUC-4 Proceedings. 1992. *Proceedings of the Fourth Message Understanding Conference*. Morgan Kaufmann, San Mateo, CA.

E. Riloff and J. Shepherd. 1997. A corpus-based approach for building semantic lexicons. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 127–132.

H. Schütze. 1992. Word sense disambiguation with sublexical representation. In *Workshop Notes, Statistically-Based NLP Techniques*, pages 109–113. AAAI.

D. Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196.

A Descriptive Characterization of Tree-Adjoining Languages (Project Note)

James Rogers
Dept. of Computer Science
Univ. of Central Florida, Orlando, FL, USA

Abstract

Since the early Sixties and Seventies it has been known that the regular and context-free languages are characterized by definability in the monadic second-order theory of certain structures. More recently, these descriptive characterizations have been used to obtain complexity results for constraint- and principle-based theories of syntax and to provide a uniform model-theoretic framework for exploring the relationship between theories expressed in disparate formal terms. These results have been limited, to an extent, by the lack of descriptive characterizations of language classes beyond the context-free. Recently, we have shown that tree-adjoining languages (in a mildly generalized form) can be characterized by recognition by automata operating on three-dimensional tree manifolds, a three-dimensional analog of trees. In this paper, we exploit these automata-theoretic results to obtain a characterization of the tree-adjoining languages by definability in the monadic second-order theory of these three-dimensional tree manifolds. This not only opens the way to extending the tools of model-theoretic syntax to the level of TALs, but provides a highly flexible mechanism for defining TAGs in terms of logical constraints.

1 Introduction

In the early Sixties Büchi (1960) and Elgot (1961) established that a set of strings was regular iff it was definable in the weak monadic second-order theory of the natural numbers with successor (wS1S). In the early Seventies an extension to the context-free languages was obtained by Thatcher and Wright (1968) and Doner (1970) who established that the CFLs were all and only the sets of strings forming the yield of sets of finite trees definable in the weak monadic second-order theory of multiple successors (wSnS). These descriptive characterizations have natural application to constraint- and principle-based theories of syntax. We have employed them in exploring the language-theoretic complexity of theories in GB (Rogers, 1994; Rogers, 1997b) and GPSG (Rogers, 1997a) and have used these model-theoretic interpretations as a uniform framework in which to compare these formalisms (Rogers, 1996). They have also provided a foundation for an approach to principle-based parsing via compilation into tree-automata (Morawietz and Cornell, 1997). Outside the realm of Computational Linguistics, these results have been employed in theorem proving with applications to program and hardware verification (Henriksen et al., 1995; Biehl et al., 1996; Kelb et al., 1997). The scope of each of these applications is limited, to some extent, by the fact that there are no such descriptive characterizations of classes of languages beyond the context-free. As a result, there has been considerable interest in extending the basic results (Mönnich, 1997; Volger, 1997) but, prior to the work reported here, the proposed extensions have not preserved the simplicity of the original results.

Recently, in (Rogers, 1997c), we introduced a class of labeled three-dimensional tree-like structures (three-dimensional tree manifolds—3-TM) which serve simultaneously as the derived and derivation structures of Tree Adjoining-Grammars (TAGs) in exactly the same way that labeled trees can serve as both derived and derivation structures for CFGs. We defined a class of automata over these structures that are a generalization of tree-automata (which are, in turn, an analogous generalization of ordinary finite-state automata over strings) and showed that the class of tree manifolds rec-