In Proceedings of the Sixth International Conference on Cognitive Modeling, 160-165.
Mahwah, NJ: Lawrence Earlbaum.

Combining learning approaches for incremental on-line parsing

Deryle Lonsdale (lonz@byu.edu)
Brigham Young University Department of Linguistics; 3186 JKHB
Provo, UT 84602 USA

Michael Manookin (mbm5@email.byu.edu)
Brigham Young University Department of Linguistics; 2129 JKHB
Provo, UT 84602 USA

Abstract

This paper discusses the integration of two different machine learning approaches to modeling language, NL-Soar and analogical modeling (AM). The resulting hybrid system is capable of functionality that is not possible when using only one of the systems in isolation. After a brief introduction of each system, an explanation is given of how AM is used to provide information useful to NL-Soar for two tasks. Examples are given, and related issues are outlined.

Introduction

Ongoing investigation in computational language modeling involves assessing the relative strengths and weaknesses of various approaches, for example symbolic versus subsymbolic, or rule-based versus exemplar-based. In this paper we show how an exemplar-based method is able to provide two types of crucial information that otherwise might not be available to a symbolic cognitive modeling system.

In the following section we sketch two substantially different machine learning approaches to language modeling. Next, we mention two well-studied natural language learning tasks: named entity recognition and prepositional phrase attachment resolution. The subsequent section discusses how two systems, NL-Soar and Analogical Modeling, have been combined in a way that brings together their relative strengths in novel and interesting ways involving these two tasks. In the last section we present conclusions, observations, and ideas for future work.

Language modeling: symbolic and exemplar-based

We begin by discussing two heretofore unrelated systems that have traditionally been used to model different language use phenomena: NL-Soar and analogical modeling (AM). Their complementarity motivates this integration: the former provides cognitive-level control, and the latter gives robust low-level instance-based matching.

Natural-language Soar

Natural-language Soar (NL-Soar) is an agent-based, hierarchical, goal-directed machine learning system that is based on the Soar cognitive modeling approach [Newell, 1990]. It has been used to model language use in a variety of modalities—comprehension [Lonsdale and Rytting, 2001], generation [Lonsdale, 2000], discourse [Green and Lehman, 2002]—in a variety of communicative task settings. A rule-based system, its basic knowledge repository is a set of if-then productions. Probabilistic reasoning is not a core feature of the basic architecture, and this introduces various challenges when addressing language-related tasks (among others).

The system receives lexical input word-by-word, and lexical access is performed for each word in turn. During lexical access WordNet [Fellbaum, 1998] provides relevant morphological, syntactic, and semantic information for all of the senses and homographs of the word in question [Rytting and Lonsdale, 2001]. The system then attempts to integrate the incoming words incrementally into linguistic models: a syntactic X-bar parse tree, and a semantic lexical-conceptual structure. All potential and possible syntactic and semantic material is considered in piecing together licit constructions. Constraints operate to rule out attachments that do not follow standard principles. In certain cases, some types of limited structure can be undone and reformulated when ongoing hypotheses prove untenable in the presence of new incoming words.

The semantic conceptual primitives are based on WordNet's lexical filenames and senses, which constitute (respectively) course-grained and fine-grained categories such as `v-body` for body verbs (e.g. sneezed, tripped) and `n-plant` for rhododendron[1]. Given the high degree of lexically-based ambiguity in English, much processing in NL-Soar involves determining compatibility between words and phrases at the syntactic and semantic levels.

The system, which is symbolic in its functionality, relies on a set of hand-coded rules and on-line learning as it performs the task. Its limited backtracking abilities allow modeling such processes as local ambiguity resolution, syntactic garden pathing, and complexity-induced breakdown in human parsing performance [Lewis, 1993].

Analogical modeling

Analogical modeling (AM) is a data-driven, exemplar-based approach to modeling language [Skousen, 1989] and other types of data. It has no rule-based component, either explicit or implicit, requires no explicit knowledge representations beyond the set of exemplars, and is a flexible and robust language modeling paradigm. Several linguistic applications have been reported using analogical modeling as the basic approach involving phonology, morphology, word sense disambiguation, speech processing, and lexical selection. So far

[1] In reality the WordNet codes are `verb.body` and `noun.plant` but this paper uses abbreviated names as shown.

it has only modeled low-level individual tasks, precluding its use as a comprehensive modeling framework.

The system operates as follows. A set of exemplars that address and illustrate a particular linguistic phenomenon is prepared; each instance has a fixed-length feature-vector encoding that represents salient (and perhaps nonsalient or questionable) properties for that instance. Each instance is labelled with an outcome that is used by the system to output how that instance behaves with respect to the phenomenon in question. At run time, the user inputs to the system a set of queries in the form of similarly encoded feature vectors. The system matches each input query with the exemplar base, and generates one or more probabilistically weighted outcomes. The system is able to tolerate noisy or incomplete data and behaves differently than other approaches in that it takes into consideration so-called "gang effects" that are problematic for the more traditional machine-learning languge-modeling methods. More details are available elsewhere concerning the system's application to language [Skousen, 1989], its statistical foundations and processing metrics [Skousen, 1992], NLP applications [Jones, 1996], and recent comparative work [Skousen et al., 2002].

Relevant tasks

In developing and scaling up the NL-Soar system, several issues arose that were not solvable using traditional symbolic methods. In this section we survey two such problems and how solutions were achieved from recent research in the natural language learning community.

Proper-noun semantics

Proper nouns are a crucial component of natural language, but in previous versions of NL-Soar they were not addressed. Using WordNet as the primary lexical resource allows for the retrieval of some of the more common proper nouns that are contained therein, such as "Virginia" and "France". For such words WordNet also provides semantic information; for example, different senses of "Washington" are encoded as `n-group`, `n-location`, and `n-person`.

Of course, only a small fraction of proper nouns are included in WordNet, which is problematic for NL-Soar's processing. A simple approach for syntax is to assume that any non-sentence-initial capitalized word encountered in text and absent from WordNet acts syntactically a proper noun[2]. Following this assumption has been straightforward and successful for handling the syntax of proper nouns. On the other hand, determining the semantics of proper nouns not included in WordNet has been more problematic.

Fortunately, this so-called named entity recognition (NER) problem has recently undergone extensive study by the natural language learning community [Tjong Kim Sang and De Meulder, 2003]. Though it has not been discussed in previous conferences on the topic, AM, like other modeling approaches, has been successfully used for named entity recognition. Using standardized data sets released from previous CoNLL shared tasks[3], AM researchers have been able to achieve state-of-the-art results for English, Dutch, and Spanish. Exemplar vector encodings used such features as the lexical item itself, its part-of-speech information, shallow-parsed constituent information, and the standard I0B codes for semantic classification.

Providing NER data for the semantics of noun phrases has been one motivation for integrating NL-Soar and AM processing. Before discussing and exemplifying this integration, we first mention another ideal application where a hybrid approach is advantageous.

PP attachment

The prepositional phrase attachment (PP-attachment) problem is an important and widely studied issue in natural language processing. Determining syntactic PP-attachment is even problematic for humans, as different attachment sites lead to multiple semantic interpretations (e.g. "I saw the man with the telescope."). Psycholinguistic research shows that human strategies for resolving PP-attachment ambiguities include the use of argument relations in the sentence [Schelstraete, 1996, Schuetze and Gibson, 1999], prosodic cues [Schafer, 1998, Straub, 1998], lexical constraints [Boland and Boehm-Jernigan, 1998], and context [Ferstl, 1994]. Others [Spivey-Knowlton and Sedivy, 1995] have demonstrated that lexical bias and contextual information have a strong effect. NL-Soar, as a rule-based symbolic system, has traditionally inferred PP-attachments primarily using lexical subcategorization information (i.e. WordNet verb frames). Thus leveraging the complement/adjunct distinction is based largely on data provided by WordNet.

Figure 1 reflects parses of two sentences: (a) "The minister warned the president of the danger.", and (b)-(c) "The minister warned the president of the republic of the danger." During parsing of the latter sentence, "...of the republic" is first temporarily linked as the PP-complement of "warned" (cf. "...of the danger" in (a)). When the second preposition is encountered, though, NL-Soar removes "...of the republic" from the verb's complement position and remakes the structure by adjoining the PP to the noun "president". The second "of" is then linked in as the PP-complement of "warned" (b). The parse then completes (c).

Though the system has been capable of handling relatively complex constructions like the one just discussed, a large class of PP-attachment scenarios could not be processed by the system. In particular, problems arose when the attachment decision was determined, not by subcategorization information on the matrix verb, but rather via lexical semantic information contained in the oblique object of the preposition. This leads to familiar structural ambiguities, which are sometimes ambiguous even for humans:

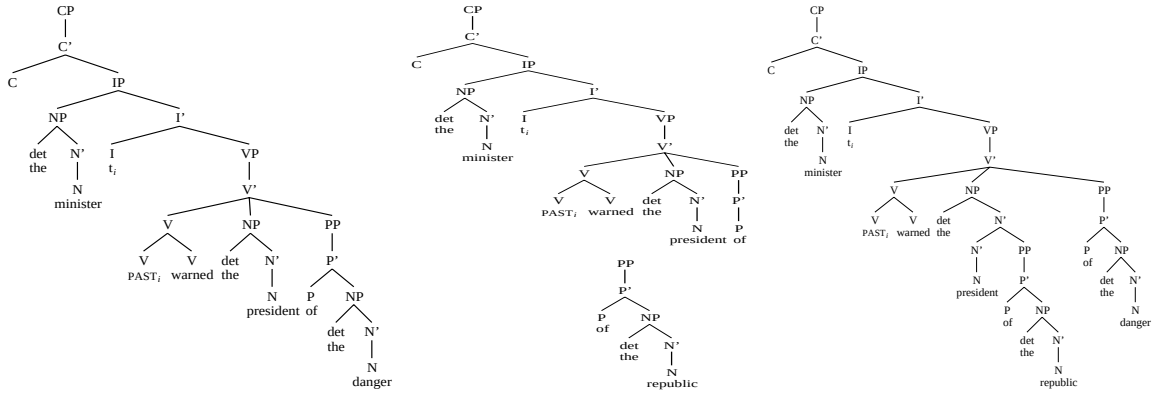
I saw the man with a beard/telescope.

Here attachment is determined by the PP object ("beard" or "telescope"), not by the subcategorization of "saw". Hence subcategorization information alone is insufficient to make PP-attachment decisions in many contexts; another approach was needed by NL-Soar to deal with ambiguities of this type. The next section presents the solution to this problem.

Past approaches to PP-attachment disambiguation have focused on statistical [Hindle and Rooth, 1993, Ratnaparkhi et al., 1994, Collins and Brooks, 1995] or rule-based [Brill and Resnik, 1994] methods. The statistical

[2]For the purposes of this paper we do not discuss words like "eBay", or "and" in multi-word proper-noun expressions.

[3]See http://lcg-www.uia.ac.be/conll200{2,3}/ner.



(a) (b) (c)

Figure 1: Sentence showing subcategorized PP complement 1(a); similar sentence with second preposition after reanalysis of the first PP now detached 1(b); final two-PP sentence including reattachment of the first PP 1(c).

approaches generally involve mining large annotated corpora and treebanks for determining the probability of an unknown attachment, usually based on the environment's lexical content. For "I saw the man with the telescope," a stochastic parser might assert an 84% probability for attachment to the verb "saw" and 16% probability for attachment to "man".

Since, for both of these tasks, exemplar-based methods are more appropriate, we describe their integration within the previously described symbolic framework.

Integration: Hybrid on-line processing

Having surveyed the NL-Soar and AM systems, we are in a position to follow how their integration allows for treatment of the two questions raised above. From a technical standpoint, composing the systems was straightforward; AM is written in C and Perl, and NL-Soar is an agent-based system written in C with a Tcl interface. It is therefore possible to call the AM system from NL-Soar via Tcl.

In the rest of this section more detail is provided about each of the integrations, and how hybrid processing unfolds.

Proper-noun semantics

Consider how NL-Soar processes the sentence: "Pendleton homered in the third inning."[4] The word "Pendleton" is not found in WordNet, and since it is also capitalized the system assumes that it is a proper noun. This initiates projection to an NP in the syntax. In order to ascertain the semantics of the word the system, during lexical access, calls the AM system with a named-entity query. The data consists simply of the word, putative part-of-speech category, and associated constituency information (i.e. initial part of an NP).

The AM system is invoked, calling the NER task with one test item—the word "Pendleton" and its associated features. The instance base consists of over 200,000 exemplars from the CoNLL-2003 shared task English NER training data. Two results are returned from the system: I-PER and I-LOC representing a person and a location, respectively. These codes are are translated into corresponding WordNet semantic classes: `n-person` and `n-location`. Syntactic and semantic nodes are created; since this is the first word of the sentence, nothing more can be done at this point.

Then the next word, "homered", arrives into the system and is attended to. This is clearly a verb, so the system's syntactic processing component projects a VP and an IP and then links in the NP "Pendleton" into the subject position. Semantically, this word is a competition verb (i.e. the semclass `v-competition` is provided by WordNet). NL-Soar, via a set of corpus-mined selectional restrictions [Rytting, 2000], is able to determine that agents of such verbs are much more likely to be of class `n-person` than of class `n-location`: people rather than places typically perform competitive actions[5]. Figure 2 shows the incremental syntactic (a) and semantic (b) representations for the sentence after the second word has been processed. Note the the LCS in (b) shows a `v-competition` node with an external argument consisting of the `n-person` construal of the proper noun; the other meaning `n-location` remains unlinked and will eventually disappear.

Thus we see that calling an exemplar-driven, corpus-based AM task for crucial NER information is useful in providing semantic information for NL-Soar. To be sure, other resources could also be accessed to find such information (e.g. online gazeteers or onomasticons), but the exemplar-based approach provides a level of generalization and robustness for processing novel items that a direct-lookup method cannot assure.

PP attachment

In similar fashion, we address the PP-attachment problem by combining NL-Soar with analogical modeling. NL-Soar calls

[4]This means that a baseball player with the surname Pendleton hit a home run in the third inning of a baseball game.

[5]Teams referred to as locations in a figurative sense may also serve as the agent in such verbs, as in "France won the World Cup."; consideration of this type of non-literal construction cannot be addressed within the scope of this paper.

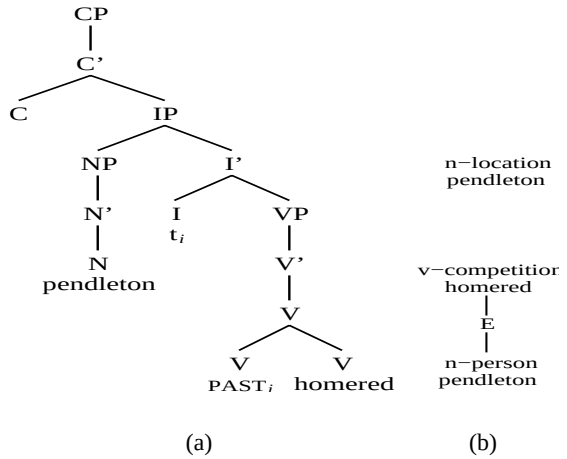


Figure 2: Syntactic and semantic incremental parses after processing the second word of the sentence.

AM when a preposition is encountered after a transitive verb and passes the verb, direct object noun, and preposition to AM. AM, in turn, runs the test item against a set of exemplars and returns the result to NL-Soar: whether the preposition should attach to the verb or to the noun.

Exemplars for this task are taken from the Brill/Resnik PP-attachment corpus[6] against an analogical set (of 12,266 examples) mined from the Penn Treebank.

Consider the sentence: "The military protects the regime against resistance." NL-Soar incrementally parses the sentence until the preposition "against" arrives; at this point there is no straightforward way for the system to decide the attachment of "against." By default, when a transitive verb in WordNet (e.g. "protect") does not subcategorize for a prepositional phrase complement, NL-Soar will adjoin the incoming PP to the direct object NP. Consequently in our sentence the PP "against resistance" will adjoin to "regime," not to "protects" (contra expectations).

In the NL-Soar/AM integrated framework, though, when "against" is encountered, a system callout queries AM for attachment preference information: the collocation "protect regime against" is sent to AM as a test item. AM computes the degree of (dis)agreement between the test item and exemplars from the previously mentioned PP-attachment corpus. The binary decision regarding the proposed attachment is returned via Tcl to NL-Soar; if licensed, the suggested attachment is implemented. If a verbal attachment is preferred, the proposed adjunction of the PP to the noun is rejected, and the preposition is instead adjoined to the verb.

Figures 3 and 4 demonstrate that integrating AM with NL-Soar generates different PP-attachments for the syntactically similar sentences "The military protects the regime against resistance." and "The military protects communication between foreign diplomats." Classified as unproblematic for humans, this type of structural ambiguity occasionally involves minor local readjustment of the parse and is relatively straightforward [Pritchett, 1992].

Figure 3(a) illustrates some of the processing involved during the parsing that creates 3(b). During cycles 259-263 NL-Soar attempts to link "against" as a complement of "protects", but fails since subcategorization requirements do not permit this attachment. Next (cycles 265-268) NL-Soar attempts to adjoin "against" to "regime"; this action is blocked by AM in cycle 268. Finally, the system attempts to adjoin "against" to "protects"; this action passes all constraints and the link is built (cycles 270-275).

A similar sequence of events occurs in Figure 4(a). The preposition "between" fails as a complement of "protects", but unlike in the previous example AM licenses[7] adjunction of "between" to "communication". Hence the link is built (cycles 253-260). The final result is shown in 4(b).

A final observation concerning learning should be made at this point. NL-Soar, through its hierarchical subgoaling architecture, learns as it performs tasks. This is true for the hybrid system described here: the system, once it has learned an attachment decision or constraint, chunks up this knowledge and retains it for future use. Thus, for example, when running the sentence in Figure 3(b) recognitionally (i.e. with all learned chunks available) vs. deliberately (i.e. with none available), the system uses one-third the CPU time (7 sec. vs. 2.3 sec.), one-half the decision cycles (500 vs. 250), one-eighth the rule firings (20,500 vs. 2500), one-tenth the number working memory changes (47,000 vs. 4700), and almost one-half the working memory size (3500 vs. 1900).

Discussion and conclusions

The work described in this paper is subject to the same criticisms that are often leveled at hybrid systems: functionality is gained, but only at the cost of abandoning a pure theoretical basis. On the other hand, much research has focused on the hypothesis that cognition (including the language faculty) could in fact be hybrid in nature. Another consideration when evaluating this tradeoff is the purpose for the system: whereas in the past NL-Soar has been used for cognitive modeling only, these problems have been encountered while trying to scale up coverage for more general NLP applications.

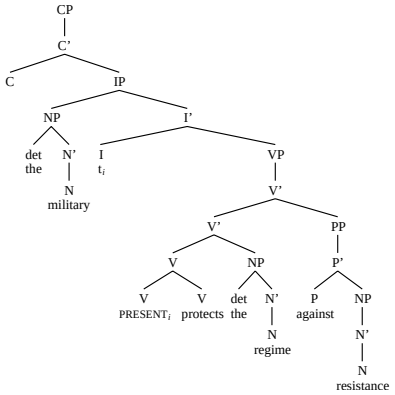
While only two exemplar-based callouts have been presented here, we anticipate that others will be useful in the future. The complement/adjunct distinction extends beyond PP's, and the determination of where to attach more complicated structures such as untensed clauses might profit from corpus-based exemplar approaches. Another difficult problem for on-line parsing is disambiguating gerund/participle environments: "My favorite cousin was singing." versus "My favorite activity was singing." Whereas this sentence pair is relatively straightforward, many cases are not as clear.

The NER data used in this paper comes off-the-shelf from a previous shared task training corpus, so the available information was limited to its contents. Only three categories (LOC, PER, ORG) have been implemented, so more thorough coverage of the semantics of proper nouns (e.g. determining gender

[6]The corpus is freely available at the following website: www.mit.edu/afs/athena/course/6/6.863/Lovecraft/PP.

[7]Since the actual exemplar is in the instance base, AM's score is 100%. Even if the instance is not used, AM approves this linkage by a margin of 72.6% to 27.4%.

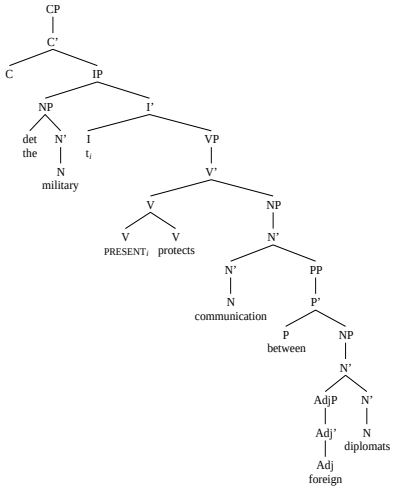
```
250: O: O221 (u-constructor46)
251: ==>S: S196 (operator no-change)
252:    ==>S: S197 (state no-change)
259:       O: O227 try(link(protects.v--comp-->against.p)
260:       ==>S: S199 (operator no-change)
261:          O: C514 (check form)
262:          O: C512 (check subcat-feature)
263:          O: C510 (check subcat)
264:       O: O231 try(link(regime.n--adjoin-->against.p)
265:       ==>S: S200 (operator no-change)
267:          O: C529 (check subcat-bead)
268:          O: C527 (check amppattach)
269:       O: O229 try(link(protects.v--adjoin-->against.p)
270:       ==>S: S201 (operator no-change)
272:          O: C538 (check subcat-bead)
273:          O: C536 (check receiver-follows)
274:          O: O237 (constraint-success)
275:    O: C487 link(protects.v--adjoin-->against.p)
276:    ==>S: S202 (state no-change)
277:       O: O239 (exhausted)
278: O: O241 (return-operator)
```



(a) (b)

Figure 3: Trace of processing during call to AM (a) and the resulting syntactic structure (b). Processing cycles in (a) are indicated on the left; note that in cycle 268 a call to AM (i.e. amppattach) blocks adjunction of the preposition to the noun.

```
238:   O: O211 (u-constructor86)
239:   ==>S: S199 (operator no-change)
240:      ==>S: S200 (state no-change)
247:         O: O217 try(link(protects.v--comp-->between.p)
248:         ==>S: S202 (operator no-change)
249:            O: C513 (check form)
250:            O: C511 (check subcat-feature)
251:            O: C509 (check subcat)
252:         O: O222 try(link(communication.n--adjoin-->between.p)
253:         ==>S: S203 (operator no-change)
255:            O: C528 (check subcat-bead)
256:            O: C526 (check amppattach)
257:            O: C524 (check verb-saturated)
258:            O: C522 (check receiver-follows)
259:            O: O227 (constraint-success)
260:      O: C488 link(communication.n--adjoin-->between.p)
261:      ==>S: S204 (state no-change)
262:         O: O229 (exhausted)
263:   O: O231 (return-operator)
```



(a) (b)

Figure 4: Trace of processing during call to AM (a) and the resulting syntactic structure (b). Processing cycles in (a) are indicated on the left; note that in cycle 256 a call to AM permits adjunction of the preposition to the noun.

of people from their names) might require more systematic corpus processing and exemplar base development.

Though empirical and theoretical issues remain, the integration of NL-Soar and AM has demonstrated how exemplar-based input can be used by a symbolic system in determining constraints, whether syntactic or semantic, for linguistic model construction. This in turn obviates the need for developing complex rule sets that likely cannot describe some phenomena in as perspicuous and robust a way as a well-constructed exemplar base.

References

[Boland and Boehm-Jernigan, 1998] Boland, J. and Boehm-Jernigan, H. (1998). Lexical constraints and prepositional phrase attachment. *Journal of Memory and Language*, 39(4):684–719.

[Brill and Resnik, 1994] Brill, E. and Resnik, P. (1994). A transformation-based approach to prepositional phrase attachment disambiguation. In *Proceedings of the Fifteenth International Conference on Computational Linguistics (COLING-1994)*.

[Collins and Brooks, 1995] Collins, M. and Brooks, J. (1995). Prepositional phrase attachment through a backed-off model. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 27–38, Cambridge, MA.

[Fellbaum, 1998] Fellbaum, C. (1998). *WordNet: An electronic lexical database*. MIT Press, Cambridge, MA.

[Ferstl, 1994] Ferstl, E. (1994). Context effects in syntactic ambiguity resolution: The location of prepositional phrase attachment. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, pages 295–300. Lawrence Erlbaum Associates.

[Green and Lehman, 2002] Green, N. and Lehman, J. F. (2002). An integrated discourse recipe-based model for task-oriented dialogue. *Discourse Processes*, 33(2).

[Hindle and Rooth, 1993] Hindle, D. and Rooth, M. (1993). Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1):103–120.

[Jones, 1996] Jones, D. (1996). *Analogical Natural Language Processing*. Studies in Computational Linguistics. University College of London Press Limited, London, England.

[Lewis, 1993] Lewis, R. (1993). An architecturally-based theory of human sentence comprehension. Technical Report CMU-CS-93-226, CMU.

[Lonsdale, 2000] Lonsdale, D. (2000). Leveraging analysis operators in incremental generation. In *Analysis for Generation: Proceedings of a Workshop at the First International Natural Language Generation Conference*, pages 9–13. Association for Computational Linguistics.

[Lonsdale and Rytting, 2001] Lonsdale, D. and Rytting, C. A. (2001). An operator-based account of semantic processing. In *The Acquisition and Representation of Word Meaning*, pages 84–92. European Summer School for Logic, Language, and Information.

[Newell, 1990] Newell, A. (1990). *Unified Theories of Cognition*. Harvard University Press.

[Pritchett, 1992] Pritchett, B. (1992). *Grammatical Competence and Parsing Performance*. University of Chicago Press.

[Ratnaparkhi et al., 1994] Ratnaparkhi, A., Reynar, J., and Roukos, S. (1994). A maximum entropy model for prepositional phrase attachment. In *Proceedings of the Human Language Technology Workshop*, pages 250–255, Plainsboro, N.J. ARPA.

[Rytting, 2000] Rytting, C. A. (2000). Semantic class disambiguation in Natural Language Soar. Brigham Young University Department of Linguistics. Unpublished honors thesis.

[Rytting and Lonsdale, 2001] Rytting, C. A. and Lonsdale, D. (2001). Integrating WordNet with NL-Soar. In *WordNet and other lexical resources: Applications, extensions, and customizations*, pages 162–164. North American Association for Computational Linguistics.

[Schafer, 1998] Schafer, A. (1998). *Prosodic parsing: The role of prosody in sentence comprehension*. PhD thesis, University of Massachusetts, Amherst.

[Schelstraete, 1996] Schelstraete, M. (1996). Definiteness and prepositional phrase attachment in sentence processing. *Current Psychology of Cognition*, 15(5):463–486.

[Schuetze and Gibson, 1999] Schuetze, C. and Gibson, E. (1999). Argumenthood and English prepositional phrase attachment. *Journal of Memory and Language*, 40(3):409–431.

[Skousen, 1989] Skousen, R. (1989). *Analogical modeling of language*. Kluwer, Dordrecht.

[Skousen, 1992] Skousen, R. (1992). *Analogy and structure*. Kluwer, Dordrecht.

[Skousen et al., 2002] Skousen, R., Lonsdale, D., and Parkinson, D., editors (2002). *Analogical modeling: An exemplar-based approach to language*. John Benjamins, Amsterdam.

[Spivey-Knowlton and Sedivy, 1995] Spivey-Knowlton, M. and Sedivy, J. (1995). Resolving attachment ambiguities with multiple constraints. *Cognition*, 55(3):227–267.

[Straub, 1998] Straub, K. (1998). *The production of prosodic cues and their role in the comprehension of syntactically ambiguous sentences*. PhD thesis, University of Rochester.

[Tjong Kim Sang and De Meulder, 2003] Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Daelemans, W. and Osborne, M., editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada.