Automatic Measurement of Syntactic Development in Child Language

Kenji Sagae and **Alon Lavie**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15232
{sagae,alavie}@cs.cmu.edu

Brian MacWhinney
Department of Psychology
Carnegie Mellon University
Pittsburgh, PA 15232
macw@cmu.edu

Abstract

To facilitate the use of syntactic information in the study of child language acquisition, a coding scheme for Grammatical Relations (GRs) in transcripts of parent-child dialogs has been proposed by Sagae, MacWhinney and Lavie (2004). We discuss the use of current NLP techniques to produce the GRs in this annotation scheme. By using a statistical parser (Charniak, 2000) and memory-based learning tools for classification (Daelemans et al., 2004), we obtain high precision and recall of several GRs. We demonstrate the usefulness of this approach by performing automatic measurements of syntactic development with the Index of Productive Syntax (Scarborough, 1990) at similar levels to what child language researchers compute manually.

1 Introduction

Automatic syntactic analysis of natural language has benefited greatly from statistical and corpus-based approaches in the past decade. The availability of syntactically annotated data has fueled the development of high quality statistical parsers, which have had a large impact in several areas of human language technologies. Similarly, in the study of child language, the availability of large amounts of electronically accessible empirical data in the form of child language transcripts has been shifting much of the research effort towards a corpus-based mentality. However, child language researchers have only recently begun to utilize modern NLP techniques for syntactic analysis. Although it is now common for researchers to rely on automatic morphosyntactic analyses of transcripts to obtain part-of-speech and morphological analyses, their use of syntactic parsing is rare.

Sagae, MacWhinney and Lavie (2004) have proposed a syntactic annotation scheme for the CHILDES database (MacWhinney, 2000), which contains hundreds of megabytes of transcript data and has been used in over 1,500 studies in child language acquisition and developmental language disorders. This annotation scheme focuses on syntactic structures of particular importance in the study of child language. In this paper, we describe the use of existing NLP tools to parse child language transcripts and produce automatically annotated data in the format of the scheme of Sagae et al. We also validate the usefulness of the annotation scheme and our analysis system by applying them towards the practical task of measuring syntactic development in children according to the Index of Productive Syntax, or IPSyn (Scarborough, 1990), which requires syntactic analysis of text and has traditionally been computed manually. Results obtained with current NLP technology are close to what is expected of human performance in IPSyn computations, but there is still room for improvement.

2 The Index of Productive Syntax (IPSyn)

The Index of Productive Syntax (Scarborough, 1990) is a measure of development of child language that provides a numerical score for grammatical complexity. IPSyn was designed for investigating individual differences in child language acqui-

Proceedings of the 43rd Annual Meeting of the ACL, pages 197–204,
Ann Arbor, June 2005.

sition, and has been used in numerous studies. It addresses weaknesses in the widely popular Mean Length of Utterance measure, or MLU, with respect to the assessment of development of syntax in children. Because it addresses syntactic structures directly, it has gained popularity in the study of grammatical aspects of child language learning in both research and clinical settings.

After about age 3 (Klee and Fitzgerald, 1985), MLU starts to reach ceiling and fails to properly distinguish between children at different levels of syntactic ability. For these purposes, and because of its higher content validity, IPSyn scores often tells us more than MLU scores. However, the MLU holds the advantage of being far easier to compute. Relatively accurate automated methods for computing the MLU for child language transcripts have been available for several years (MacWhinney, 2000).

Calculation of IPSyn scores requires a corpus of 100 transcribed child utterances, and the identification of 56 specific language structures in each utterance. These structures are counted and used to compute numeric scores for the corpus in four categories (noun phrases, verb phrases, questions and negations, and sentence structures), according to a fixed score sheet. Each structure in the four categories receives a score of zero (if the structure was not found in the corpus), one (if it was found once in the corpus), or two (if it was found two or more times). The scores in each category are added, and the four category scores are added into a final IPSyn score, ranging from zero to 112.[1]

Some of the language structures required in the computation of IPSyn scores (such as the presence of auxiliaries or modals) can be recognized with the use of existing child language analysis tools, such as the morphological analyzer MOR (MacWhinney, 2000) and the part-of-speech tagger POST (Parisse and Le Normand, 2000). However, more complex structures in IPSyn require syntactic analysis that goes beyond what POS taggers can provide. Examples of such structures include the presence of an inverted copula or auxiliary in a wh-question, conjoined clauses, bitransitive predicates, and fronted or center-embedded subordinate clauses.

[1]See (Scarborough, 1990) for a complete listing of targeted structures and the IPSyn score sheet used for calculation of scores.

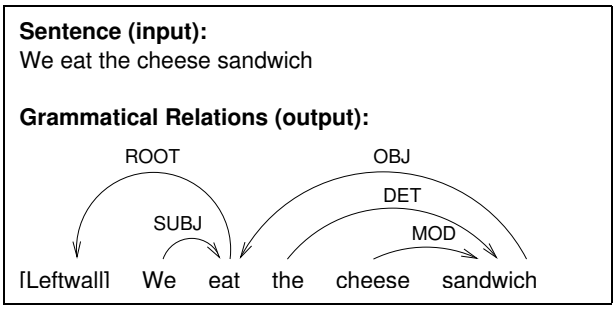

Figure 1: Input sentence and output produced by our system.

3 Automatic Syntactic Analysis of Child Language Transcripts

A necessary step in the automatic computation of IPSyn scores is to produce an automatic syntactic analysis of the transcripts being scored. We have developed a system that parses transcribed child utterances and identifies grammatical relations (GRs) according to the CHILDES syntactic annotation scheme (Sagae et al., 2004). This annotation scheme was designed specifically for child-parent dialogs, and we have found it suitable for the identification of the syntactic structures necessary in the computation of IPSyn.

Our syntactic analysis system takes a sentence and produces a labeled dependency structure representing its grammatical relations. An example of the input and output associated with our system can be seen in figure 1. The specific GRs identified by the system are listed in figure 2.

The three main steps in our GR analysis are: text preprocessing, unlabeled dependency identification, and dependency labeling. In the following subsections, we examine each of them in more detail.

3.1 Text Preprocessing

The CHAT transcription system[2] is the format followed by all transcript data in the CHILDES database, and it is the input format we use for syntactic analysis. CHAT specifies ways of transcribing extra-grammatical material such as disfluency, retracing, and repetition, common in spontaneous spoken language. Transcripts of child language may contain a large amount of extra-grammatical mate-

[2]http://childes.psy.cmu.edu/manuals/CHAT.pdf

SUBJ, ESUBJ, CSUBJ, XSUBJ Subject, expletive subject, clausal subject (finite and non–finite)			**OBJ, OBJ2, IOBJ** Object, second object, indirect object		
COMP, XCOMP Clausal complement (finite and non–finite)			**PRED, CPRED, XPRED** Predicative, clausal predicative (finite and non–finite)		
JCT, CJCT, XJCT Adjunct, clausal adjunct (finite and non–finite)			**MOD, CMOD, XMOD** Nominal modifier, clausal nominal modifier (finite and non–finite)		
AUX Auxiliary	**NEG** Negation	**DET** Determiner	**QUANT** Quantifier	**POBJ** Prepositional object	**PTL** Verb particle
CPZR Complementizer	**COM** Communicator	**INF** Infinitival "to"	**VOC** Vocative	**COORD** Coordinated item	**ROOT** Top node

Figure 2: Grammatical relations in the CHILDES syntactic annotation scheme.

rial that falls outside of the scope of the syntactic annotation system and our GR identifier, since it is already clearly marked in CHAT transcripts. By using the CLAN tools (MacWhinney, 2000), designed to process transcripts in CHAT format, we remove disfluencies, retracings and repetitions from each sentence. Furthermore, we run each sentence through the MOR morphological analyzer (MacWhinney, 2000) and the POST part-of-speech tagger (Parisse and Le Normand, 2000). This results in fairly clean sentences, accompanied by full morphological and part-of-speech analyses.

3.2 Unlabeled Dependency Identification

Once we have isolated the text that should be analyzed in each sentence, we parse it to obtain unlabeled dependencies. Although we ultimately need labeled dependencies, our choice to produce unlabeled structures first (and label them in a later step) is motivated by available resources. Unlabeled dependencies can be readily obtained by processing constituent trees, such as those in the Penn Treebank (Marcus et al., 1993), with a set of rules to determine the lexical heads of constituents. This lexicalization procedure is commonly used in statistical parsing (Collins, 1996) and produces a dependency tree. This dependency extraction procedure from constituent trees gives us a straightforward way to obtain unlabeled dependencies: use an existing statistical parser (Charniak, 2000) trained on the Penn Treebank to produce constituent trees, and extract unlabeled dependencies using the aforementioned head-finding rules.

Our target data (transcribed child language) is from a very different domain than the one of the data used to train the statistical parser (the Wall Street Journal section of the Penn Treebank), but the degradation in the parser's accuracy is acceptable. An evaluation using 2,018 words of in-domain manually annotated dependencies shows that the dependency accuracy of the parser is 90.1% on child language transcripts (compared to over 92% on section 23 of the Wall Street Journal portion of the Penn Treebank). Despite the many differences with respect to the domain of the training data, our domain features sentences that are much shorter (and therefore easier to parse) than those found in Wall Street Journal articles. The average sentence length varies from transcript to transcript, because of factors such as the age and verbal ability of the child, but it is usually less than 15 words.

3.3 Dependency Labeling

After obtaining unlabeled dependencies as described above, we proceed to label those dependencies with the GR labels listed in Figure 2.

Determining the labels of dependencies is in general an easier task than finding unlabeled dependencies in text.[3] Using a classifier, we can choose one of the 30 possible GR labels for each dependency, given a set of features derived from the dependencies. Although we need manually labeled data to train the classifier for labeling dependencies, the size of this training set is far smaller than what would be necessary to train a parser to find labeled dependen-

[3]Klein and Manning (2002) offer an informal argument that constituent labels are much more easily separable in multidimensional space than constituents/distituents. The same argument applies to dependencies and their labels.

cies in one pass.

We use a corpus of about 5,000 words with manually labeled dependencies to train TiMBL (Daelemans et al., 2003), a memory-based learner (set to use the k-nn algorithm with k=1, and gain ratio weighing), to classify each dependency with a GR label. We extract the following features for each dependency:

- The head and dependent words;
- The head and dependent parts-of-speech;
- Whether the dependent comes before or after the head in the sentence;
- How many words apart the dependent is from the head;
- The label of the lowest node in the constituent tree that includes both the head and dependent.

The accuracy of the classifier in labeling dependencies is 91.4% on the same 2,018 words used to evaluate unlabeled accuracy. There is no intersection between the 5,000 words used for training and the 2,018-word test set. Features were tuned on a separate development set of 582 words.

When we combine the unlabeled dependencies obtained with the Charniak parser (and head-finding rules) and the labels obtained with the classifier, overall labeled dependency accuracy is 86.9%, significantly above the results reported (80%) by Sagae et al. (2004) on very similar data.

Certain frequent and easily identifiable GRs, such as DET, POBJ, INF, and NEG were identified with precision and recall above 98%. Among the most difficult GRs to identify were clausal complements COMP and XCOMP, which together amount to less than 4% of the GRs seen the training and test sets. Table 1 shows the precision and recall of GRs of particular interest.

Although not directly comparable, our results are in agreement with state-of-the-art results for other labeled dependency and GR parsers. Nivre (2004) reports a labeled (GR) dependency accuracy of 84.4% on modified Penn Treebank data. Briscoe and Carroll (2002) achieve a 76.5% F-score on a very rich set of GRs in the more heterogeneous and challenging Susanne corpus. Lin (1998) evaluates his MINIPAR system at 83% F-score on identification of GRs, also in data from the Susanne corpus (but using simpler GR set than Briscoe and Carroll).

GR	Precision	Recall	F-score
SUBJ	0.94	0.93	0.93
OBJ	0.83	0.91	0.87
COORD	0.68	0.85	0.75
JCT	0.91	0.82	0.86
MOD	0.79	0.92	0.85
PRED	0.80	0.83	0.81
ROOT	0.91	0.92	0.91
COMP	0.60	0.50	0.54
XCOMP	0.58	0.64	0.61

Table 1: Precision, recall and F-score (harmonic mean) of selected Grammatical Relations.

4 Automating IPSyn

Calculating IPSyn scores manually is a laborious process that involves identifying 56 syntactic structures (or their absence) in a transcript of 100 child utterances. Currently, researchers work with a partially automated process by using transcripts in electronic format and spreadsheets. However, the actual identification of syntactic structures, which accounts for most of the time spent on calculating IPSyn scores, still has to be done manually.

By using part-of-speech and morphological analysis tools, it is possible to narrow down the number of sentences where certain structures may be found. The search for such sentences involves patterns of words and parts-of-speech (POS). Some structures, such as the presence of determiner-noun or determiner-adjective-noun sequences, can be easily identified through the use of simple patterns. Other structures, such as front or center-embedded clauses, pose a greater challenge. Not only are patterns for such structures difficult to craft, they are also usually inaccurate. Patterns that are too general result in too many sentences to be manually examined, but more restrictive patterns may miss sentences where the structures are present, making their identification highly unlikely. Without more syntactic analysis, automatic searching for structures in IPSyn is limited, and computation of IPSyn scores still requires a great deal of manual inspection.

Long, Fey and Channell (2004) have developed a software package, Computerized Profiling (CP), for child language study, which includes a (mostly)

automated computation of IPSyn.[4] CP is an extensively developed example of what can be achieved using only POS and morphological analysis. It does well on identifying items in IPSyn categories that do not require deeper syntactic analysis. However, the accuracy of overall scores is not high enough to be considered reliable in practical usage, in particular for older children, whose utterances are longer and more sophisticated syntactically. In practice, researchers usually employ CP as a first pass, and manually correct the automatic output. Section 5 presents an evaluation of the CP version of IPSyn.

Syntactic analysis of transcripts as described in section 3 allows us to go a step further, fully automating IPSyn computations and obtaining a level of reliability comparable to that of human scoring. The ability to search for both grammatical relations and parts-of-speech makes searching both easier and more reliable. As an example, consider the following sentences (keeping in mind that there are no explicit commas in spoken language):

(a) Then [,] he said he ate.
(b) Before [,] he said he ate.
(c) Before he ate [,] he ran.

Sentences (a) and (b) are similar, but (c) is different. If we were looking for a fronted subordinate clause, only (c) would be a match. However, each one of the sentences has an identical part-speech-sequence. If this were an isolated situation, we might attempt to fix it by having tags that explicitly mark verbs that take clausal complements, or by adding lexical constraints to a search over part-of-speech patterns. However, even by modifying this simple example slightly, we find more problems:

(d) Before [,] he told the man he was cold.
(e) Before he told the story [,] he was cold.

Once again, sentences (d) and (e) have identical part-of-speech sequences, but only sentence (e) features a fronted subordinate clause. These limited toy examples only scratch the surface of the difficulties in identifying syntactic structures without syntactic analysis beyond part-of-speech and morphological tagging. In these sentences, searching with GRs is easy: we simply find a GR of clausal type (e.g. CJCT, COMP, CMOD, etc) where the dependent is to the left of its head.

For illustration purposes of how searching for structures in IPSyn is done with GRs, let us look at how to find other IPSyn structures[5]:

- Wh-embedded clauses: search for wh-words whose head, or transitive head (its head's head, or head's head's head...) is a dependent in GR of types [XC]SUBJ, [XC]PRED, [XC]JCT, [XC]MOD, COMP or XCOMP;
- Relative clauses: search for a CMOD where the dependent is to the right of the head;
- Bitransitive predicate: search for a word that is a head of both OBJ and OBJ2 relations.

Although there is still room for under- and over-generalization with search patterns involving GRs, finding appropriate ways to search is often made trivial, or at least much more simple and reliable than searching without GRs. An evaluation of our automated version of IPSyn, which searches for IPSyn structures using POS, morphology and GR information, and a comparison to the CP implementation, which uses only POS and morphology information, is presented in section 5.

5 Evaluation

We evaluate our implementation of IPSyn in two ways. The first is *Point Difference*, which is calculated by taking the (unsigned) difference between scores obtained manually and automatically. The point difference is of great practical value, since it shows exactly how close automatically produced scores are to manually produced scores. The second is *Point-to-Point Accuracy*, which reflects the overall reliability over each individual scoring decision in the computation of IPSyn scores. It is calculated by counting how many decisions (identification of presence/absence of language structures in the transcript being scored) were made correctly, and dividing that

[4] Although CP requires that a few decisions be made manually, such as the disambiguation of the lexical item "'s" as copula vs. genitive case marker, and the definition of sentence breaks for long utterances, the computation of IPSyn scores is automated to a large extent.

[5] More detailed descriptions and examples of each structure are found in (Scarborough, 1990), and are omitted here for space considerations, since the short descriptions are fairly self-explanatory.

number by the total number of decisions. The point-to-point measure is commonly used for assessing the inter-rater reliability of metrics such as the IPSyn. In our case, it allows us to establish the reliability of automatically computed scores against human scoring.

5.1 Test Data

We obtained two sets of transcripts with corresponding IPSyn scoring (total scores, and each individual decision) from two different child language research groups. The first set (A) contains 20 transcripts of children of ages ranging between two and three. The second set (B) contains 25 transcripts of children of ages ranging between eight and nine.

Each transcript in set A was scored fully manually. Researchers looked for each language structure in the IPSyn scoring guide, and recorded its presence in a spreadsheet. In set B, scoring was done in a two-stage process. In the first stage, each transcript was scored automatically by CP. In the second stage, researchers checked each automatic decision made by CP, and corrected any errors manually.

Two transcripts in each set were held out for development and debugging. The final test sets contained: (A) 18 transcripts with a total of 11,704 words and a mean length of utterance of 2.9, and (B) 23 transcripts with a total of 40,819 words and a mean length of utterance of 7.0.

5.2 Results

Scores computed automatically from transcripts parsed as described in section 3 were very close to the scores computed manually. Table 2 shows a summary of the results, according to our two evaluation metrics. Our system is labeled as GR, and manually computed scores are labeled as HUMAN. For comparison purposes, we also show the results of running Long et al.'s automated version of IPSyn, labeled as CP, on the same transcripts.

Point Difference

The average (absolute) point difference between automatically computed scores (GR) and manually computed scores (HUMAN) was 3.3 (the range of HUMAN scores on the data was 21-91). There was no clear trend on whether the difference was positive or negative. In some cases, the automated scores were higher, in other cases lower. The minimum difference was zero, and the maximum difference was 12. Only two scores differed by 10 or more, and 17 scores differed by two or less. The average point difference between HUMAN and the scores obtained with Long et al.'s CP was 8.3. The minimum was zero and the maximum was 21. Sixteen scores differed by 10 or more, and six scores differed by 2 or less. Figure 3 shows the point differences between GR and HUMAN, and CP and HUMAN.

It is interesting to note that the average point differences between GR and HUMAN were similar on sets A and B (3.7 and 2.9, respectively). Despite the difference in age ranges, the two averages were less than one point apart. On the other hand, the average difference between CP and HUMAN was 6.2 on set A, and 10.2 on set B. The larger difference reflects CP's difficulty in scoring transcripts of older children, whose sentences are more syntactically complex, using only POS analysis.

System	Avg. Pt. Difference to HUMAN	Point-to-Point Reliability
GR (Total)	**3.3**	**92.8%**
CP (Total)	8.3	85.4%
GR (Set A)	3.7	92.5%
CP (Set A)	6.2	86.2%
GR (Set B)	2.9	93.0%
CP (Set B)	10.2	84.8%

Table 2: Summary of evaluation results. GR is our implementation of IPSyn based on grammatical relations, CP is Long et al.'s (2004) implementation of IPSyn, and HUMAN is manual scoring.

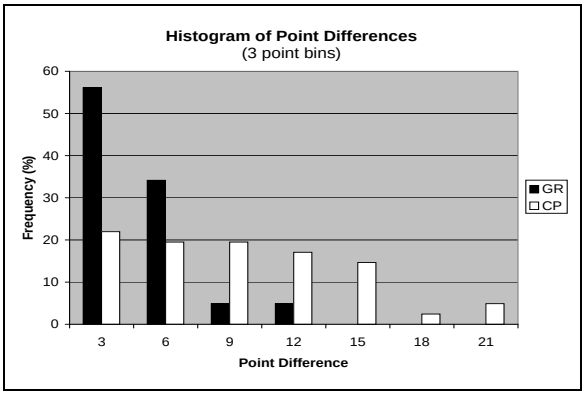

Figure 3: Histogram of point differences between HUMAN scores and GR (black), and CP (white).

Point-to-Point Accuracy

In the original IPSyn reliability study (Scarborough, 1990), point-to-point measurements using 75 transcripts showed the mean inter-rater agreement for IPSyn among human scorers at 94%, with a minimum agreement of 90% of all decisions within a transcript. The lowest agreement between HUMAN and GR scoring for decisions within a transcript was 88.5%, with a mean of 92.8% over the 41 transcripts used in our evaluation. Although comparisons of agreement figures obtained with different sets of transcripts are somewhat coarse-grained, given the variations within children, human scorers and transcript quality, our results are very satisfactory. For direct comparison purposes using the same data, the mean point-to-point accuracy of CP was 85.4% (a relative increase of about 100% in error).

In their separate evaluation of CP, using 30 samples of typically developing children, Long and Channell (2001) found a 90.7% point-to-point accuracy between fully automatic and manually corrected IPSyn scores.[6] However, Long and Channell compared only CP output with manually corrected CP output, while our set A was manually scored from scratch. Furthermore, our set B contained only transcripts from significantly older children (as in our evaluation, Long and Channell observed decreased accuracy of CP's IPSyn with more complex language usage). These differences, and the expected variation from using different transcripts from different sources, account for the difference in our results and Long and Channell's.

5.3 Error Analysis

Although the overall accuracy of our automatically computed scores is in large part comparable to manual IPSyn scoring (and significantly better than the only option currently available for automatic scoring), our system suffers from visible deficiencies in the identification of certain structures within IPSyn.

Four of the 56 structures in IPSyn account for almost half of the number of errors made by our system. Table 3 lists these IPSyn items, with their respective percentages of the total number of errors.

[6]Long and Channell's evaluation also included samples from children with language disorders. Their 30 samples of typically developing children (with a mean age of 5) are more directly comparable to the data used in our evaluation.

IPSyn item	Error
S11 (propositional complement)	16.9%
V15 (copula, modal or aux for emphasis or ellipsis)	12.3%
S16 (relative clause)	10.6%
S14 (bitransitive predicate)	5.8%

Table 3: IPSyn structures where errors occur most frequently, and their percentages of the total number of errors over 41 transcripts.

Errors in items S11 (propositional complements), S16 (relative clauses), and S14 (bitransitive predicates) are caused by erroneous syntactic analyses. For an example of how GR assignments affect IPSyn scoring, let us consider item S11. Searching for the relation COMP is a crucial part in finding propositional complements. However, COMP is one of the GRs that can be identified the least reliably in our set (precision of 0.6 and recall of 0.5, see table 1). As described in section 2, IPSyn requires that we credit zero points to item S11 for no occurrences of propositional complements, one point for a single occurrence, and two points for two or more occurrences. If there are several COMPs in the transcript, we should find about half of them (plus others, in error), and correctly arrive at a credit of two points. However, if there are very few or none, our count is likely to be incorrect.

Most errors in item V15 (emphasis or ellipsis) were caused not by incorrect GR assignments, but by imperfect search patterns. The searching failed to account for a number of configurations of GRs, POS tags and words that indicate that emphasis or ellipsis exists. This reveals another general source of error in our IPSyn implementation: the search patterns that use GR analyzed text to make the actual IPSyn scoring decisions. Although our patterns are far more reliable than what we could expect from POS tags and words alone, these are still hand-crafted rules that need to be debugged and perfected over time. This was the first evaluation of our system, and only a handful of transcripts were used during development. We expect that once child language researchers have had the opportunity to use the system in practical settings, their feedback will allow us to refine the search patterns at a more rapid pace.

6 Conclusion and Future Work

We have presented an automatic way to annotate transcripts of child language with the CHILDES syntactic annotation scheme. By using existing resources and a small amount of annotated data, we achieved state-of-the-art accuracy levels.

GR identification was then used to automate the computation of IPSyn scores to measure grammatical development in children. The reliability of our automatic IPSyn was very close to the inter-rater reliability among human scorers, and far higher than that of the only other computational implementation of IPSyn. This demonstrates the value of automatic GR assignment to child language research.

From the analysis in section 5.3, it is clear that the identification of certain GRs needs to be made more accurately. We intend to annotate more in-domain training data for GR labeling, and we are currently investigating the use of other applicable GR parsing techniques.

Finally, IPSyn score calculation could be made more accurate with the knowledge of the expected levels of precision and recall of automatic assignment of specific GRs. It is our intuition that in a number of cases it would be preferable to trade recall for precision. We are currently working on a framework for soft-labeling of GRs, which will allow us to manipulate the precision/recall trade-off as discussed in (Carroll and Briscoe, 2002).

Acknowledgments

This work was supported in part by the National Science Foundation under grant IIS-0414630.

References

Edward J. Briscoe and John A. Carroll. 2002. Robust accurate statistical annotation of general text. *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, (pp. 1499–1504). Las Palmas, Gran Canaria.

John A. Carroll and Edward J. Briscoe. 2002. High precision extraction of grammatical relations. *Proceedings of the 19th International Conference on Computational Linguistics*, (pp. 134-140). Taipei, Taiwan.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. *In Proceedings of the First Annual Meeting of the North American Chapter of the Association for Computational Linguistics.* Seattle, WA.

Michael Collins. 1996. A new statistical parser based on bigram lexical dependencies. *Proceedings of the 34th Meeting of the Association for Computational Linguistics* (pp. 184-191). Santa Cruz, CA.

Walter Daelemans, Jacub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2004. TiMBL: Tilburg Memory Based Learner, version 5.1, Reference Guide. *ILK Research Group Technical Report Series* no. 04-02, 2004.

T. Klee and M. D. Fitzgerald. 1985. The relation between grammatical development and mean length of utterance in morphemes. *Journal of Child Language*, 12, 251-269.

Dan Klein and Christopher D. Manning. 2002. A generative constituent-context model for improved grammar induction. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (pp. 128-135).

Dekang Lin. 1998. Dependency-based evaluation of MINIPAR. *In Proceedings of the Workshop on the Evaluation of Parsing Systems.* Granada, Spain.

Steve H. Long and Ron W. Channell. 2001. Accuracy of four language analysis procedures performed automatically. *American Journal of Speech-Language Pathology*, 10(2).

Steven H. Long, Marc E. Fey, and Ron W. Channell. 2004. Computerized Profiling (Version 9.6.0). Cleveland, OH: Case Western Reserve University.

Brian MacWhinney. 2000. The CHILDES Project: Tools for Analyzing Talk. Mahwah, NJ: Lawrence Erlbaum Associates.

Mitchel P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewics. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19.

Joakim Nivre and Mario Scholz. 2004. Deterministic dependency parsing of English text. *Proceedings of International Conference on Computational Linguistics* (pp. 64-70). Geneva, Switzerland.

Christophe Parisse and Marie-Thrse Le Normand. 2000. Automatic disambiguation of the morphosyntax in spoken language corpora. *Behavior Research Methods, Instruments, and Computers*, 32, 468-481.

Kenji Sagae, Alon Lavie, and Brian MacWhinney. 2004. Adding Syntactic annotations to transcripts of parent-child dialogs. *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*. Lisbon, Portugal.

Hollis S. Scarborough. 1990. Index of Productive Syntax. *In Applied Psycholinguistics*, 11, 1-22.