DOI 10.3758/s13428-013-0354-x

# Automatic generation of the index of productive syntax for child language transcripts

**Khairun-nisa Hassanali · Yang Liu · Aquiles Iglesias · Thamar Solorio · Christine Dollaghan**

Published online: 30 May 2013

**Abstract** The index of productive syntax (IPSyn; Scarborough (Applied Psycholinguistics 11:1–22, 1990) is a measure of syntactic development in child language that has been used in research and clinical settings to investigate the grammatical development of various groups of children. However, IPSyn is mostly calculated manually, which is an extremely laborious process. In this article, we describe the AC-IPSyn system, which automatically calculates the IPSyn score for child language transcripts using natural language processing techniques. Our results show that the AC-IPSyn system performs at levels comparable to scores computed manually. The AC-IPSyn system can be downloaded from www.hlt.utdallas.edu/~nisa/ipsyn.html.

**Keywords** Automatic computation · Child language · Index of productive syntax · Language measures · Natural language processing

## Introduction

There has been a long-standing interest in the development of metrics for quantifying the structural aspects of expressive language in children and adults (e.g., developmental sentence scoring (DSS), Lee, 1974; developmental level (D-level), Covington, He, Brown, Naci, & Brown, 2006; Rosenberg & Abbeduto, 1987; quantitative analysis of agrammatic production, Saffran, Berndt, & Schwartz, 1989; and the index of productive syntax [IPSyn], Scarborough, 1990). As was noted by Hughes, Fey, and Long (1992) in their discussion of the DSS, the advantage of these analytical approaches is that they provide a single numeric score that can be used to compare performance across individuals and to establish developmental norms that can be used to compare an individual to group performance. Although these single numeric scores do not fully capture the varied and complex nature of the language sample, they provide researchers and clinicians with an overall index of performance.

Time constraints have been a major roadblock in the utilization of several of these analytical approaches. Manual analysis of language samples is extremely laborious and requires skilled analysts to identify the relevant syntactic constructs in the sample. In part due to the time-consuming nature of manual analysis, the DSS measure requires the user to examine only the first 50 utterances, whereas the IPSyn measure requires only the first 100 utterances. The use of natural language processing techniques for fully automatic morpho-syntactic analysis of language samples has the potential of greatly enhancing the ability of researchers and practitioners to take advantage of the information available in samples of spontaneous language.

IPSyn (Scarborough, 1990) is one analytical approach that has maintained a considerable popularity over the last decades, and one for which automated systems exist (e.g., the Computerized Profiling software, Long, Fey, & Channell, 2004; or the Sagae system, Sagae, Lavie, & MacWhinney, 2005). The IPSyn has been used with a wide array of ages, dialects, languages, and disorders (Geers,

K.-n. Hassanali (✉) · Y. Liu
Computer Science Department, University of Texas at Dallas,
Richardson 75080 TX, USA
e-mail: nisa@hlt.utdallas.edu

A. Iglesias
Department of Communication Sciences and Disorders,
Temple University, Philadelphia, PA, USA

T. Solorio
Department of Computer and Information Sciences,
University of Alabama at Birmingham, Birmingham, AL, USA

C. Dollaghan
Callier Center for Communication Disorders,
University of Texas at Dallas, Richardson, TX, USA

2002; Hadley, 1998; Hewitt, Hammer, Yont, & Tomblin, 2005; Nieminen, 2009; Oetting et al., 2010). The IPSyn considers structures in four major syntactic categories: noun phrase (NP; adjectives, modifiers, nouns, plural nouns, and two- and three-word NPs), verb phrase (VP; different verb forms, adverbs, and VPs), questions and negations (intonational questions, wh-questions and negations), and sentences (syntactic constructs that look at later-developing syntactic abilities such as the use of relative clauses, passive constructs, and tag questions). In total, 60 grammatical structures (12 nouns, 17 verbs, 11 questions and negations, and 20 sentences) are assessed. A given construct can receive 0 points (never occurs), 1 point (occurs once in the sample), or 2 points (occurs twice or more). Since the IPSyn is designed to measure the emergence of particular grammatical forms, two unique occurrences of a construct are considered sufficient. This simple scoring procedure gives the IPSyn an advantage over the DSS, which requires each utterance to be independently scored for a variety of constructs.

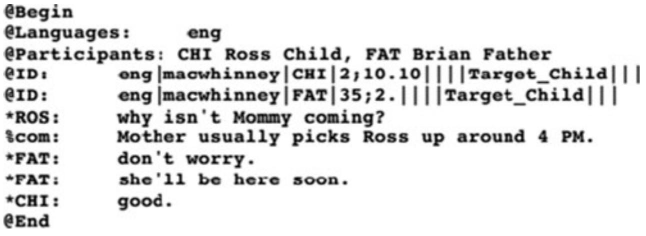
Recent advances in the area of natural language processing have resulted in the development of automated systems that can calculate the IPSyn with little or no human input. The Computerized Profiling system (CP) provides automated computation of the IPSyn score, as well as of other measures including developmental level. The CP software uses morphological analysis and part-of-speech (POS) tagging, but no syntactic parser, to calculate the IPSyn score. Although mostly automated, the CP requires some manual input, such as distinguishing between the possessive and the copula (e.g., Joe's shoe vs. Joe's here). Sagae et al. (2005) suggested that the CP system is not ideal for analyzing older children with more complex syntax because of its inability to identify IPSyn categories that require deep syntactic analysis. They further suggested that CP is currently used as a first pass, with subsequent manual correction of the output. Using CP outputs results in significant time savings, since verifying the correctness of the constructs extracted by CP is faster than manual identification of each of the syntactic constructs.

The Sagae system's approach to IPSyn, by contrast, is fully automated, taking sentences as input and extracting predefined grammatical relations from the parse trees for each sentence using the Charniak (2000) parser and memory-based tools. These grammatical relations include subject, object, complementizer, and negation relations. An IPSyn score is computed using grammatical relations to identify occurrences of IPSyn syntactic constructs. The Sagae system is reportedly more accurate than CP (Sagae et al., 2005), identifying 92.5 % of all structures identified by human annotators. Although an improvement over the CP, the Sagae system is not as precise as could be desired. Furthermore, the Sagae system is not publicly available to the research and clinical community, limiting its practical utility.

The aim of our project was to develop a fully automatic system to produce the IPSyn that could process several hundreds of transcripts at a time with reasonable accuracy and that would be freely available to the research community. In the system development phase, we wanted the system to analyze each of the identified IPSyn constructs along with scores across any range of utterances. One of the motivations behind developing our system was to provide additional insight into the features that contribute to language acquisition and learning. Keeping this in mind, our system was designed with the option of extracting the count of occurrences for each grammatical structure and syntactic category identified in the IPSyn. This feature allows researchers to analyze the presence and degree of productivity of each item assessed. The system also enables researchers to rapidly access individual occurrences of constructs of interest for more detailed analyses. Finally, we wanted to ensure that our system was able to take as input language samples transcribed using either the CHAT (MacWhinney, 2000) or SALT (Miller & Iglesias, 2008) formats. The CHAT and SALT transcript formats allow users to mark extra information such as errors and disfluencies. The CHAT manual can be downloaded at http://childes.psy.cmu.edu/manuals/CHAT.pdf, and a summary of the SALT transcription format can be downloaded from www.saltsoftware.com/salt/TranConvSummary.pdf. Listings 1 and 2 give examples of CHAT and SALT transcripts, respectively.

## Development of the Automatic Computation of IPSyn system (AC-IPSyn)

The development of the Automatic Computation of IPSyn system (AC-IPSyn) involved four distinct steps: preprocessing, parsing, identification of IPSyn structures, and the computation of scores. It should be noted that, in addition to the overall IPSyn score, the system was developed to provide: (a) a list of each occurrence



```
@Begin
@Languages:     eng
@Participants: CHI Ross Child, FAT Brian Father
@ID:        eng|macwhinney|CHI|2;10.10||||Target_Child|||
@ID:        eng|macwhinney|FAT|35;2.||||Target_Child|||
*ROS:       why isn't Mommy coming?
%com:       Mother usually picks Ross up around 4 PM.
*FAT:       don't worry.
*FAT:       she'll be here soon.
*CHI:       good.
@End
```

**Listing 1** Sample CHAT transcript

```
$ CHILD, EXAMINER
+ Name: Andy
+ Gender: M
+ CA: 6;7
+ Grade: 1st
+ Context: NAR
- 0:00

C (Uh) Everybody/'s up.
E Mhm.
C They/'re go/ing (up and st*) up and down the stairs, up and down the
stairs.
E Good.
C There was[EW:were] alot of kid/s.
C And the police officer in the house they say "What/'s go/ing on" [EU]?
C (D*) and he was say/ing "Do your parent/s live here"?
C "Yeah, they/'re here now".
C "But they don't live here".
E Um.
C And (the uh) the pizza came.
C And (h*) the (uh) Home_Alone kid (:03 he uh he was he was) he was look/ing
for cheese pizza.
C And he could/n't find any.
C (Uh) his brother eated|eat[EO:ate] it.
C And he said "Get a plate".
C "I/'m gonna barf".
C And (he) he was get/ing mad.
C And he hitted|hit[EO:hit] X tummy, hid all the milk [EU].
C And he (had) go/3s up to third (fo*) floor.
E Mhm.
C And (uh) *the next day (he said he uh) he sleeped|sleep[EO:slept] for that
day <> up in the attic.
```

**Listing 2** Sample SALT transcript

of an IPSyn syntactic construct, indexed by the line number where it first appeared in the transcript, along with the points scored on every syntactic construct; and (b) a summary of the scores in each of the four IPSyn categories (nouns, verbs, questions, and sentences) and subcategories.

*Step 1. Preprocessing*

The first step in the process is to ensure that the transcript is in an appropriate format. Both CHAT and SALT format are accepted. Each utterance is segmented, and false starts are identified. Transcripts are then stripped of any transcription conventions (e.g., "/", used in SALT to separate bound and unbound morphemes), and codes and contractions are converted to full forms.

*Step 2. Parsing*

After the preprocessing step, a syntactic analysis of the transcript is performed using the Charniak (2000) parser. The Charniak parser first assigns POS tags to each word in each sentence of the transcript and then uses these POS tags to generate a syntactic analysis of the utterance. For example, the sentence "She is a girl" would be tagged as "She (PRONOUN) is (COPULA) a (ARTICLE) girl (NOUN)." After POS tagging, the Charniak parser generates a syntactic analysis (i.e., the parse tree) of the sentence. For example, the utterance "It kind of looks like it's a something." is parsed by the Charniak parser as follows: (S1 (S (NP (PRP It)) (ADVP (RB kind) (IN of)) (VP (VBZ looks) (SBAR (IN like) (S (NP (PRP it)) (VP (AUX 's)[1] (NP (DT a) (NN something)))))) (.)).[2]

In natural languages, there are various ambiguities; a word may have multiple possible POS tags, and a sentence may have multiple parses. The key issue in POS tagging and syntactic parsing is thus to resolve such ambiguities. This can often be done using context information. In the past decades, many statistical approaches have been developed for these tasks, achieving reasonably good performance. In these methods, annotated corpora (data labeled with POS tags and parse trees) are used to train statistical models. During testing, these models determine the most likely analysis for the given sentence. The Charniak (2000) parser, which is the parser used in Sagae et al. (2005) and the present study, has a reported precision/recall averages of

[1] The Charniak parser tags both the copula and auxiliary verbs as AUX. We check for the presence of an extra verb to distinguish between copula and auxiliary verbs.

[2] The tags mark each verb phrase (VP), noun phrase (NP), adverbial phrase (ADVP), personal pronoun (PRP), determiner (DT), adverb (RB), singular noun or mass noun (NN), preposition or a subordinating conjunction (IN), auxiliary verb or copula (AUX), third-person singular present tense verb (VBZ), simple declarative clause (S), clause introduced by a subordinating conjunction (SBAR), and sentence terminator ("."). These tags follow the Penn Treebank annotation. For more information, refer to Taylor, Marcus, and Santorini (2003).

90.1 % for sentences of maximum length 40 words and 89.5 % for sentences of maximum length 100 words (Charniak, 2000) on *Wall Street Journal* data. According to Sagae et al., although the Charniak parser has been trained on adult language, it performs reasonably well on child language samples. This seemed to be the case in our manual examination of the parsed trees: The majority of parsing errors observed were due to the parser encountering words such as "oops" that were prevalent in child language but not present in the corpus on which the Charniak parser was trained. It should be noted that the parsing errors do not impact the system's performance significantly, since the IPSyn scoring requires only two exemplars of a construct, and most of the constructs, when present, had numerous exemplars.

### *Step 3. Identifying IPSyn structures*

Rules were created to identify each of the IPSyn syntactic constructs from the POS tags and the constituent parse trees.[3] Our system differs from that of Sagae et al. (2005) in that we did not use a corpus to train a classifier that detects relations in sentences (subject, object, etc.). Instead, we constructed rules based directly on the POS tagging and parsing results of the transcripts to detect the syntactic constructs. For the constructs that just required POS tags, regular expressions that search for a particular POS tag were constructed. For example, when searching for utterances that contain either the gerund or a progressive, the rule identified all utterances with a "VBG" tag and searched the context to distinguish whether the word was a progressive or a gerund. For some constructs, rules were applied to the parsed trees. The system traverses the trees to identify the constituent subtrees, which consist of a root node and an ordered list of its immediate children. For example, to identify wh-questions with an inverted modal, copula, or auxiliary, the rule was to search for a subtree with the head SBARQ (i.e., direct question introduced by a wh-word or wh-phrase) that further had a subtree with the head SQ (i.e., inverted yes–no question or main clause of a wh-question following the wh-phrase in SBARQ).

### *Step 4. Computation of the scores*

Once the occurrences of all IPSyn structures were identified, the system calculated the score for each grammatical structure, the total score for each of the four syntactic categories examined, and the total score. In deriving specific scores, the system takes into account Scarborough's (1990) guidelines for exceptions and constraints of uniqueness. For example, when searching for exemplars of three-word noun phrases, the guidelines suggest that at least two of the three words should differ in two exemplars for them to be considered as productive word combinations, rather than memorized or "frozen" forms. Also, nouns that are normally used in their plural form (e.g., *pants*) were not considered plural forms.

## Evaluation of the AC-IPSyn

To evaluate our AC-IPSyn system, we use two data sets. Data Set A corresponded to Set A used by Sagae et al. (2005),[4] which consisted of 20 transcripts from typically developing (TD) children between 2 and 3 years of age with an average mean length of utterance in morphemes (MLUm) of 2.9. This set contained a total of 11,704 words. Data Set B comprised 20 transcripts selected from among 677 transcripts collected from 6-year-old children in the course of a study of the relation of otitis media and child development (Paradise et al., 2005). As was reported by Gabani, Solorio, Liu, Hassanali, and Dollaghan (2011), 623 of the 677 transcripts were labeled as TD and 54 as language impaired (LI). For Data Set B, ten transcripts of each type were selected at random. Data Set B contained 10,254 words, with an average MLUm of 3.5.

For the purpose of system development, we randomly selected five additional transcripts of TD children from the Paradise data set (Paradise et al., 2005) to tune the rules used in the AC-IPSyn system. These transcripts were not included in Data Set B and did not contribute to system evaluation, however.

Consistent with the procedures described in Sagae et al. (2005), system performance was evaluated using two measures, point difference and point-to-point accuracy. These are calculated by comparing the system scores to manual IPSyn scoring of the transcripts in each dataset. The *point difference* is the absolute difference between the IPSyn total points scores computed manually and automatically; its potential range was 0 to 120. This measure shows how close the automatically computed scores are to the manual scores. *Point-to-point accuracy* captures the agreement between the manual identification and the system's identification of the presence or absence of individual IPSyn syntactic constructs. It is calculated by counting the number of agreements between the manual identification and the system identification for each of the 60 grammatical structures and the sum divided by the total number of decisions.

[3] Note that for the N12 category defined in IPSyn (i.e., constructs that have not been seen in categories N1 to N11; Scarborough, 1990), no rules were defined, since we did not find such examples in our data.

[4] We thank Kenji Sagae at the University of Southern California for generously providing this data set.

**Table 1** Average point difference and point-to-point accuracy between manual scoring and the three systems (AC-IPSyn, Sagae, and CP)

| Data Set | System | Avg. Point Difference | Point-to-Point Accuracy (%) |
|---|---|---|---|
| Data Set A | AC-IPSyn | 3.05 (*SD* 2.14) | 96.9 |
| | Sagae | 3.7 | 92.5 |
| | CP | 8.3 | 86.2 |
| Data Set B | AC-IPSyn | 3.05 | 96.4 |
| | Sagae | n/a | n/a |
| | CP | 6.55 | 87.39 |

Table 1 shows the scores for our AC-IPSyn system, and the CP and Sagae systems. It should be noted that the Sagae system is not available, and the results presented for that system are based on Sagae et al.'s (2005) reported values. Thus, no results are available on how the Sagae system would perform on Data Set B.

As can be seen in Table 1, the average point differences between manual scoring and scoring by the AC-IPSyn and the Sagae systems for Data Set A were 3.05 and 3.7, respectively. Sagae et al. (2005) had reported that the average point difference for CP was 8.3 for this data set. For Data Set B, the AC-IPSyn outperformed the CP system (3.05 vs. 6.55, respectively). With respect to point-to-point accuracy, the AC-IPSyn (96.2 %) outperformed the Sagae (92.5 %) and CP (86.2 %) systems for Data Set A. In addition, the AC-IPSyn outperformed the CP system for Data Set B (96.4 % vs. 87.39 %, respectively). On the basis of the average point difference and point-to-point accuracy, the results from the AC-IPSyn were more similar to the results from manual scoring than were those of either the CP software or the approach described by Sagae et al.

The differences between our system and the system developed by Sagae et al. (2005) might be a result of the more robust rules and patterns that we developed. Additionally, because the performance of Sagae et al.'s system was dependent on the grammatical relations extracted using classification, any error in classifying of grammatical relations would be propagated to errors in identifying of IPSyn syntactic constructs. CP's relatively poorer performance can be attributed to the fact that CP uses only POS tagging and morphological analysis, and thus would have more difficulty identifying the sentence constructs. We also observed more errors in the CP software's POS tagging; for example, verbs such as *see* and *do* were identified as nouns for all of the transcripts that we examined. These errors have an impact on the computation of the IPSyn score.

The AC-IPSyn system performed relatively better on transcripts that had multiple occurrences of a syntactic construct. In this case, even if AC-IPSyn failed to identify one of the syntactic constructs, the correct identification of the other constructs would result in a correct IPSyn score. As one would expect, error rates tend to be higher if the transcript has only a single occurrence of the construct. An analysis of the errors in the AC-IPSyn output suggested that most were due to incorrect POS tagging and parsing. Another source of error in our system was due to exemplars not matching the regular expressions that we constructed for the rules. For example, the rule for S12 (conjoined sentences) expects one conjunction between the two sentences. However, we had an instance in which a child used the conjunction *and* twice between the sentences, resulting in the software missing the instance of S12.

## Using the AC-IPSyn system

The AC-IPSyn system is a Linux/UNIX-based command line system. If Linux is not installed on the machine, users could run Linux from a USB or DVD, or use a virtual Linux machine. Users need to have installed the Python and Perl packages in addition to the AC-IPSyn package; both of the former packages are freely available. The Charniak parser and Tree tagger (Schmid, 1997) software used by the AC-IPSyn system is provided in the AC-IPSyn package. The program takes as input transcripts in the CHAT or SALT format. A user could provide as input a single transcript or a directory containing multiple transcripts. A user needs to provide the code used to label the child's utterances (e.g., "CHI" in CHILDES and "C" in SALT). The system then extracts the utterances of the child and processes them. The output—containing the overall IPSyn score, the score for each of the four syntactic categories, and a listing of specific structures used to calculate each individual construct—is stored in a directory specified by the user. See the Appendix for a screenshot of the AC-IPSyn system and an example output, containing the overall and syntactic category scores as well as an example of structures used to calculate the score of a particular construct. The Linux version of the AC-IPSyn system can be downloaded from www.hlt.utdallas.edu/~nisa/ipsyn.html. A user manual with instructions on the installation and use of the AC-IPSyn system is also provided.

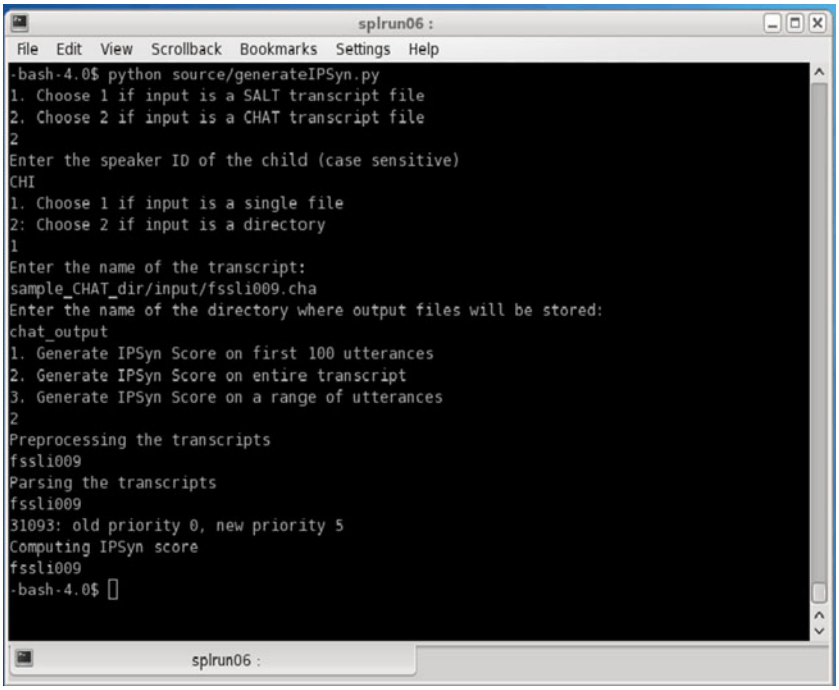
## Conclusions and future work

Manual scoring of a transcript for IPSyn takes, on average, up to half an hour to score the first 100 utterances. The AC-IPSyn system, which is fully automated and allows for batch processing, is capable of scoring 100 utterances in less than 5 min—a significant time savings. The CP system asks for manual input in order to identify the possessive form, which makes it more time consuming. Also, since CP does not support batch processing, it takes several hours to process the same number of transcripts that could be processed by the AC-IPSyn system in an hour. Furthermore, our system provides the flexibility to identify all IPSyn constructs on any range of utterances. The system also allows for extracting the exact count of the IPSyn syntactic constructs, which provides researchers with the flexibility for analyses beyond the IPSyn specifications. In the future, we plan to improve our system by formulating more robust rules and incorporating more syntactic structures, especially for identifying more complex sentence constructs, making the system more amenable for the analysis of older children.

**Author note** This research is supported in part by NSF Award Nos. IIS-1017190 and 1018124. Some of the data for these analyses were originally obtained in the course of a research project led by Jack L. Paradise, which was supported by grants from the National Institute of Child Health and Human Development, the Agency for Healthcare Research and Quality, and the National Institutes of Health General Clinical Research Center, in addition to gifts from SmithKline Beecham Laboratories and Pfizer.

## Appendix

Screenshot



```
-bash-4.0$ python source/generateIPSyn.py
1. Choose 1 if input is a SALT transcript file
2. Choose 2 if input is a CHAT transcript file
2
Enter the speaker ID of the child (case sensitive)
CHI
1. Choose 1 if input is a single file
2: Choose 2 if input is a directory
1
Enter the name of the transcript:
sample_CHAT_dir/input/fssli009.cha
Enter the name of the directory where output files will be stored:
chat_output
1. Generate IPSyn Score on first 100 utterances
2. Generate IPSyn Score on entire transcript
3. Generate IPSyn Score on a range of utterances
2
Preprocessing the transcripts
fssli009
Parsing the transcripts
fssli009
31093: old priority 0, new priority 5
Computing IPSyn score
fssli009
-bash-4.0$ 
```

**Fig. A1** Generating IPSyn scores on an entire single CHAT transcript.

Sample Results

```
###N9 Three word NP (Det-Mod-N or Mod-Mod-N)###
26    THE DIRTY SILVERWARE
29    A COUPLE MINUTES
33    ANY MORE PIECES
72    THE ONLY WAY
77    THIS YELLOW STUFF
96    THE RED CUP
100   A GREEN CUP
100   THE GREEN SILVERWARE
SCORE:     2

###V7 Progressive Suffix###
13    COOKING
36    GOING
57    GOING
62    USEING
74    CLEANING
87    MAKEING
89    EATING
SCORE:     2

###V9 Modal preceeding verb###
2     'LL THE SHELFS

26    CAN BE ALL THE DIRTY SILVERWARE
39    COULD EAT THAT
53    'LL BE RIGHT
72    CAN GO
91    'LL BE DONE IN TWO MORE
SCORE:     2

###V12     Regular past tense suffix###
82    COOLED
SCORE:     1
```

```
###Q3 Simple Negation + X ) neg = no(t), can't don't X = NP,PP,VP,Adj,
Adv etc.###
1     OH , NO ONE 'S UNDER HERE .
28    NOT YET .
46    I DO NOT KNOW .
66    OH THERE 'S NOT .
81    THAT CAR DOES NOT GO OVER THERE ! THAT MAKES IT LOOK LIKE IT 'S
SWERVY .
83    I JUST NEED PLATES AND I DO NOT HAVE ANY .
92    YES BUT THOSE DID NOT DIRTY .
SCORE:     2

###Q6 Wh-question with inverted modal, copula or auxillary###
51    WHAT 'S THIS FOR ?
68    HOW MANY DO YOU THINK ?
84    WHAT DO YOU WANT TO DRINK ?
SCORE:     2

###S12     Conjoined sentences (Except for imperatives, will usually
have subj + predicate in each clause)###
83    I JUST NEED PLATES AND I DO NOT HAVE ANY
SCORE:     1

#########################
NOUN SCORE: 20
VERB SCORE: 30
QUESTION SCORE: 14
SENTENCE SCORE: 28
IPSYN SCORE: 92
##########################
```

## References

Charniak, E. (2000). *A maximum-entropy-inspired parser. In NAACL 2000: Proceedings of the 1st North American Chapter of the Association for Computational Linguistics (pp. 132–139)*. Stroudsburg: Association for Computational Linguistics.

Covington, M. A., He, C., Brown, C., Naci, L., & Brown, J. (2006). How complex is that sentence? A proposed revision of the Rosenberg and Abbeduto D-Level scale. *Computer Analysis of Speech for PsychologicalResearch* (*CASPR*) *Research Report 2006-01*. Athens, GA: The University of Georgia, Artificial Intelligence Center. Retrieved March 20, 2013, from www.ai.uga.edu/caspr/2006-01-Covington.pdf

Gabani, K., Solorio, T., Liu, Y., Hassanali, K., & Dollaghan, C. A. (2011). Exploring a corpus-based approach for detecting language impairment in monolingual English-speaking children. *Artificial Intelligence in Medicine, 53,* 161–170. doi:10.1016/j.artmed.2011.08.001

Geers, A. (2002). Factors affecting the development of speech, language, and literacy in children with cochlear implantation. *Language Speech and Hearing Services in the Schools, 33,* 172–183. doi:10.1044/0161-1461(2002/015)

Hadley, P. A. (1998). Early verb-related vulnerability among children with specific language impairment. *Journal of Speech, Language, and Hearing Research, 41,* 1384–1397.

Hewitt, L. E., Hammer, C. S., Yont, K. M., & Tomblin, J. B. (2005). Language sampling for kindergarten children with and without SLI: Mean length of utterance, IPSYN, and NDW. *Journal of Communication Disorders, 38,* 197–213. doi:10.1016/j.jcomdis.2004.10.002

Hughes, D., Fey, M. E., & Long, S. H. (1992). Developmental sentence scoring: Still useful after all of these years. *Topics in Language Disorders, 12,* 1–12.

Lee, L. (1974). Developmental sentence analysis: A grammatical assessment procedure for speech and language clinicians. Evanston, IL: Northwestern University Press.

Long, S. H., Fey, M. E., & Channell, R. W. (2004). *Computerized Profiling (Version 9.6.0) [Computer software]*. Cleveland, OH: Case Western Reserve University.

MacWhinney, B. (2000). *The CHILDES project: Tools for analyzing talk. Vol. I: Transcription format and programs*. Mahwah: Erlbaum.

Miller, J., & Iglesias, A. (2008). *Systematic Analysis of Language Transcripts (SALT), English & Spanish (Version 9) [Computer Software]*. Madison: University of Wisconsin–Madison, Waisman Center, Language Analysis Laboratory.

Nieminen, L. (2009). MLU and IPSyn measuring absolute complexity. *Estonian Papers in Applied Linguistics, 5,* 173–185.

Oetting, J. B., Newkirk, B. L., Hartfield, L. R., Wynn, C. G., Pruitt, S. L., & Garrity, A. W. (2010). Index of productive syntax for children who speak African American English. *Language, Speech, and Hearing Services in Schools, 41,* 328–339. doi:10.1044/0161-1461(2009/08-0077)

Paradise, J. L., Campbell, T. F., Dollaghan, C. A., Feldman, H. M., Bernard, B. S., Colborn, D. K., . . . Smith, C. G. (2005). Developmental outcomes after early or delayed insertion of tympanostomy tubes. New England Journal of Medicine, 353, 576–586. doi:10.1056/NEJMoa050406

Rosenberg, S., & Abbeduto, L. (1987). Indicators of linguistic competence in the peer group conversational behavior of mildly retarded adults. *Applied Psycholinguistics, 8,* 19–32. doi:10.1017/S0142716400000047

Saffran, E. M., Berndt, R. S., & Schwartz, M. F. (1989). The quantitative analysis of agrammatic production: Procedure and data. *Brain and Language, 37,* 440–479.

Sagae, K., Lavie, A., & MacWhinney, B. (2005). Automatic measurement of syntactic development in child language. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics* (pp. 197–204). New Brunswick: Association for Computational Linguistics.

Scarborough, H. S. (1990). Index of productive syntax. *Applied Psycholinguistics, 11,* 1–22. doi:10.1017/S0142716400008262

Schmid, H. (1997). Probabilistic part-of-speech tagging using decision trees. In D. B. Jones & H. L. Somers (Eds.), *New methods in language processing* (pp. 154–164). London: UCL Press.

Taylor, A., Marcus, M., & Santorini, B. (2003). The Penn Treebank: An overview. In A. Abeillé (Ed.), *Treebanks: Building and using parsed corpora* (pp. 5–22). Dordrecht: Kluwer. doi:10.1007/978-94-010-0201-1_1