9-2016

An Evaluation of POS Taggers for the CHILDES Corpus

Rui Huang

The Graduate Center, City University of New York

How does access to this work benefit you? Let us know!

Follow this and additional works at: https://academicworks.cuny.edu/gc_etds

Part of the Computational Linguistics Commons, and the First and Second Language Acquisition Commons

Recommended Citation

Huang, Rui, "An Evaluation of POS Taggers for the CHILDES Corpus" (2016). CUNY Academic Works.
https://academicworks.cuny.edu/gc_etds/1577

This Thesis is brought to you by CUNY Academic Works. It has been accepted for inclusion in All Dissertations, Theses, and Capstone Projects by an authorized administrator of CUNY Academic Works. For more information, please contact deposit@gc.cuny.edu.
AN EVALUATION OF POS TAGGERS FOR THE
CHILDES CORPUS

by

RUI HUANG

A master’s thesis submitted to the Graduate Faculty in Linguistics in partial fulfillment of the requirements for the degree of Master of Arts, The City University of New York

2016
An Evaluation of POS Taggers for The CHILDES Corpus

by

Rui Huang

This manuscript has been read and accepted for the Graduate Faculty in Linguistics in satisfaction of the thesis requirement for the degree of Master of Arts.

Date

Advisor Name: William Gregory Sakas, PhD
Thesis Advisor

Date

EO Name: Gita Martohardjono, PhD
Executive Officer

THE CITY UNIVERSITY OF NEW YORK
ABSTRACT

An Evaluation of POS taggers for The CHILDES corpus

by

Rui Huang

Advisor: William Sakas

This project evaluates four mainstream taggers on a representative collection of child-adult’s dialogues from Child Language Data Exchange System. The nine children’s files from Valian corpora and part of Eve corpora have been manually labeled, and rewrote with LARC tagset. They served as gold standard corpora in the training and testing process. Four taggers: CLAN MOR tagger, ACOPOST trigram tagger, Stanford parser, and Ver. 1.14 of Brill tagger have been tested by 10-fold cross validation. By analyzing what kinds of assumptions the tagger made about category assignment lead to failing, we identify several problematic cases of tagging. By comparing the average error rate of each tagger, we found the size of training data set, and the length of utterance both plays a role to effect tagging accuracy.
Contents

List of Figures

List of Tables

1 Background Information

1.1 Corpus Analysis in Linguistics

1.2 Part of Speech Tagging

1.3 POS Tagging in Child’s Language

2 Corpus Construction

2.1 Data

2.2 Manual Annotation of the Corpora

3 Evaluation

3.1 Four Taggers

3.1.1 CLAN MOR Tagger

3.1.2 ACOPOST Trigram Tagger

3.1.3 Brill Tagger

3.1.4 Stanford Tagger
3.2 Evaluation Methodology and Tagging accuracy

3.3 Problematic Tagging cases

4 Qualitative Error Analysis

4.1 CLAN MOR tagging difference on Valian and Eve corpus

4.2 Tagging accuracy and the size of training data

5 Conclusion
List of Figures

1. Confusion Matrix of ACOPOST Trigram Tagging Result
2. Confusion Matrix of Brill Tagger Tagging Result
3. Confusion Matrix of Stanford Tagging Result
4. Precision and Recall of tagging result on Adult’s speech
5. Precision and Recall of tagging result on Child’s speech
6. Words per Tag Distribution of Data

List of Tables

1. List of LARC Tags
2. Corpus Utterances number and Word counts
3. Unigram, Brill, Trigram, and Stanford tagging accuracy
4. CLAN MOR tagging Accuracy
1 Background information

1.1 Corpus analysis in linguistics

The analysis of language corpora has a long history and it appears in different places around the world. For example, Panini, the ancient Indian linguist, describes the grammar of classical Sanskrit in 4th century BCE based on Vedas texts. Vedas is a large Hindu scripture, which composed in Vedic Sanskrit. Another example is the development of Arabic grammar dictionary. At the very beginning, Arabic grammarian lists language rules without any explanation. Since mid-600s, they have paid attentions to Arabic poetry and exegesis of the Qur'an. They have used these classics as corpora to explain Arabic grammar, and constructed grammar dictionary in more detail. Nevertheless, western theologians have worked for centuries to gather every edition of the canonical Bible texts and prepare concordances in order to do detailed study of the Holy Bible.

In recent decades, the widespread use of the Internet and the high-performance computational chips made it is possible to collect, store and process an astronomical mount of information. In 1967, Henry Kucera and W. Nelson Francis have first practiced their research ‘Computational Analysis of Present-Day American English’ on electronically stored and
processed texts. Those texts were adding up to one million words and saved in a unified format. It is named as “Brown Corpus” because it was assembled in Brown University.

Soon after the key publication of Brown Corpus, computational calculation of language constructions in the corpora has gradually become a major instrument to test linguistic hypothesis. Another benefit of the electronic version of data corpora is that the corpora are reusable and convenient to share. Many hands make light work; a growing number of researchers would like to contribute their data to the public, and get the access to others’ data via the World Wide Web. Collaborative research projects have shed some light on encountering the problem of luck of data corpus. For example, the open source project, such as Human Genome Project, in which biological scientists share their data with other researchers over the world, has gained a great success. As in linguistic community, The Brown Corpus has been uploaded on line for public downloading. Other corpus, such as British National Corpus, Contemporary American English Corpus, and Talk Bank, are also available or free download. People who have accessed the Internet can use it for their own research.

1. 2 Part of speech tagging
The task of classifying words in a natural language text with respect to a specific criterion is known as “tagging”. Different types of tagging can be distinguished based on the specific criterion employed. In linguistics, researchers usually tag the word with its part of speech category. A part of speech (PoS) is a category to which a word is assigned in accordance with its grammatical properties. It is also named as morphological classes, word classes, speech categories, lexical items or lexical tags. For example, *noun*, *verb*, *adverb*, *adjective*, and *preposition* are common parts of speech tags in modern English. These part of speech tags are representing the information of the word and its neighbors in a sentence.

A tagged linguistic corpus provides researchers with better means for exploring instances, finding frequencies of particular constructions and searching specific usage. Usually, it is very difficult to spot syntactic constructions in a plain text corpus. For example, the word “to” is multifunctional in the English language. It can be a marker of prepositional phrases, *to*-complement clauses, and relative clauses. If researchers want to find all the “to” that are used as a marker of prepositional phrases, they have to look at every occurrence of “to” to determine whether or not it was the target usage. When working with several large corpuses, this work would turn out to be an extremely time consuming task. However, if the
word “to” has a tag as “preposition”, this investigation might be done in one step by finding out all the “to” with a “preposition” tag. Similarly, part of speech tagging tells us how the word is pronounced and thus enhances the accuracy of speech recognition. For instance, the verb “read” is pronounced as /ri:d/ when it is used as a simple present tense verb, and it is pronounced as /red/ when it is used as a simple past tense verb. With a corpus that correctly tagged with part of speech tags, the analysis can be done relatively quickly. As of all these convenient in linguistic research, correctly tagging part of speech to the data corpora is important in computational speech and language processing.

The machine that does the computational procedures for assigning parts of speech to words is called a “part of speech tagger”. Most taggers are issued online as open source research projects, and are published for free download. The state-of-arts TreeTagger achieved 97.5% per-word accuracy on German newspaper corpus (Schimid, 1995); and 97.24% on the Penn Treebank tagged Wall Street Journal corpus (Toutanova et al., 2003). In fact, the first pass tagging accuracy of a human tagger is around 94% (MacWhiney, 2000). Compared to manual tagging result, part of speech taggers have done with an outstanding work on grammatical text, such as edited books or newspapers. However, in an evaluation of five part
of speech taggers, the accuracy dropped below 93% on informal writing, such as web page texts. (Giesbrecht et al., 2009).

1.3 POS tagging in child’s language

Most experimental participants of first language acquisition (FLA) research are infants or young children who cannot write. Spoken language data, like spontaneous speech sound, are recorded in this case. To perform statistical analysis on voice data, researchers have to transfer oral speech voice to written language text. The process of writing out spoken language data from an audio file is called ‘transcription’. The newly developed data-driven method was applied to written language corpora at first, and soon found its way to transcribe spoken language corpora. Although there are no direct comparisons of different taggers on either child or adult speech corpora; it can be expected that various transcribing conventions, unconscious repetitions in spoken language and similar appearance of different syntactical structure could make computational coding difficult. For instance, a developer reported trigram parser mistagged 7.39% of child word and 4.56% of adult words (Schroder, I. 2002). Another example is the CLAN MOR tagger, which is the most widely used children’s part of speech parser. Its developers reported accuracy on dependency parsing is 90.0% in labeled
accuracy score (LAS), and 91.7% in unlabeled accuracy score (UAS). LAS means the parser predicts the correct head-word and dependency label; while UAS means the parser corresponds to percentage of words for which the correct head was found. In another word, one out of ten pairs of dependency words of child’s speech is mis-tagged. Obviously, this tagging result on spontaneous speech corpora is insufficient for researchers who deal with direct data analysis in psycholinguistics. A hand editing process of verifying the outcomes of automatic analyses and correcting those inaccurate tags that would affect the final result is inevitable at this stage. To ensure a high accuracy of tagging result, the time consuming hand-editing process of correctly assigning tags is required.

What’s more, developments of computational methods make it possible to collect and analyze on a large language corpus. Approximately, there are 700,000 utterances from 92 cross-sectional and 638 longitudinal children and 800,000 adult utterances stored in Child Language Data Exchange System (CHILDES), which are open to public download. As parts of speech are the building blocks of grammar, the hand-editing process has prevented the use of large sampling data in linguistic research. Therefore, making an accurate and complete tagger becomes an urgent and interesting question.
This project therefore proposed to evaluate four kinds of taggers on spontaneous speech from children and adults conversation; to analyze the types of error from each tagger; and to determine whether the errors are similar for children and adults. The rest of the paper is structured as follows: Section 2 describes the corpus used in this project. Section 3 describes our research methodology in conducting the experiments. Section 4 provides a qualitative analysis of tagging results. And, section 5 suggests some insights for developing a more robust part of speech tagger, according to statistical results from the previous section.
2 Corpus construction

2.1 Data

We used nine children’s data from Valian corpus and children’s data from Eve corpus in the current study. The Valian corpus contains 1.5 hours of spontaneous speech from each of 21 children ranging in MLU from 1.53 to 4.38. It is one of the largest cross-sectional corpora in the CHILDES database. In this project, we investigated 9 out of 21 children in Valian corpus. Spontaneous speeches from these children are saved in 17 files. After excluding unintelligible utterances, like xxx, yyy and www, there are 16,945 fully hand coded utterances, 19,118 adults’ and 5,770 children’s. The utterances are 64,595 words, 51,176 adults’ and 13,419 children’s. The average mean length of utterances is 3.81, 5.27 for adults’ and 2.41 for children’s.

To complement with Valian corpus, we used 5,770 eligible children’s utterances from the Eve corpus. The Eve corpus is one of the three children from Roger Brown corpora. It is fully hand coded by researchers from Carnegie Mellon University when they are doing the CLAN project. In all 20 files of Eve corpus, there are 26,346 fully hand coded utterances, 14,807 adults’ and 12,113 children’s. The utterances are 90,859 words, 59,076 adults’ and
31,779 children’s. The average mean length of utterances is 4.5, 3.6 for children and 5.3 for adults.

2.2 Manual annotation of the corpora

The source data are served as “gold standard” in the tagging process. To make a sound experiment, we focused our manual annotation on above-mentioned corpora in this session.

The corpora are saved as CHA files in the CHILDES project. CHA files conform to a CHAT format unique to CLAN MOR tagger. Each utterance in the CHA file is written in two tiers: the speaker tier which is marked with “*”, and the MOR tier which is marked with “%”. For example:

*CHI: more cookie .

%mor: qn|more n|cookie .

The speaker tier contains the speaker’s speech transcript. The MOR tier indicates the morphological structure of each word in the speaker tier through its notation.

The self-reported accuracy rate of this coding of MOR tier is around 97%. This result is not good enough to be a gold-standard data set for the tagging experiment. So the first step in corpus construction is to find out and correct all errors that made by CLAN MOR tagger. To
ensure the accuracy of correction, we had three different people to do the task. The first person went through the transcription, found out error tags, and wrote down the possible correct tags. The second person went through all the corrections that were made by the first person, and decided whether approve the correction or not. If the first two people happened to find the same correction, this word is recorded and would be corrected in the following process. If the first two people did not agree with each other, the word would redirect to a third person, and the third person made a final decision. For some of the child’s utterances, particularly the shortest ones, is hard to be determinate. In these cases, we discussed them in our group meeting and aimed to make them into a unified tagging schema.

Specifically, we wrote a preprocessing script (tagcheker.py) prompts one utterance at a time for the error checking. All tag errors, which were found by the first person, were finalized as comma separated value (CSV) for further processing. For the approval procedure, we made a web application named as “LARC Childes Correction Approver” (http://childes-corex.pfeyz.webfactional.com) that prompted both the original utterance and the tag errors in the same web page. The second person made decisions via the same web application. Disagreements between the first person and the second person were saved in
CSV files as well, and were redirected to the third person, waiting for the final decisions.

Through its notation, MOR can indicate the morphological structure of a word. Unlike other taggers, MOR tagger does a morphological analysis before part of speech tagging. Thus CHA files conform to a format unique to CLAN (CHAT) that contains many symbols with morphological meaning. This morphological information is not acceptable by other programs. In order to compare the parsers, it is necessary to have a common set of tags that all the parsers can work with. We spent a large portion of the year converting the MOR tags to one which has both internal and external validity and can be used by all 4 parsers.

We gathered all MOR tag symbols and made an exhaustive list of MOR tag symbols. Based on the MOR tag list, we created a LARC (Language Acquisition Research Center) tag set, along with an annotation manual providing a rationale for the changes (See LARC tag manual). In many cases we accepted the MOR tags as given (e.g., adjective, past tense verb); in others we collapsed tags (e.g., all adjectives receive the same label, even if they are comparatives); in a few cases we added tags (e.g., modals) or reinstated distinctions that had been erased (e.g., preposition VS. particle) because of the importance of the distinction. By creating these common tags, we can better compare the 4 parsers.
To convert MOR tags into LARC tags, the ‘word/MOR-tag’ files need to be stripped of all CHAT symbols. In order to do this, we wrote a converting script (named translate.py in the package) that takes a CSV file representing the MOR to LARC tag transformations and applied these transformations to TXT files. From XML files, we reformatted our corpus to a ‘word/MOR-tag’ TXT file, wrote a script (apply_correction.py) to apply corrected tags to error tag, and got a corrected ‘word/MOR-tag’ TXT file. Eve corpora, which are tagged with CHAT symbols, are reformatted to LARC standards before actual training/testing process.

Based on the result from MOR tagger, we manually checked the part of speech tags of Valian corpora, and reformatted CHAT symbols to LARC standards as well.
<table>
<thead>
<tr>
<th>Table 1. List of LARC Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.ADJ: adjective</td>
</tr>
<tr>
<td>2.ADV: adverb</td>
</tr>
<tr>
<td>3.AUX: auxiliary</td>
</tr>
<tr>
<td>4.CHI: child word</td>
</tr>
<tr>
<td>5.CO0: coordinating conjuction</td>
</tr>
<tr>
<td>6.COP: copular</td>
</tr>
<tr>
<td>7.DET: determiner</td>
</tr>
<tr>
<td>8.EXT: existential there</td>
</tr>
<tr>
<td>9.FIL: filler</td>
</tr>
<tr>
<td>10.GER: gerund</td>
</tr>
<tr>
<td>11.GER-PL: plural gerund</td>
</tr>
<tr>
<td>12.IDF: indefinite pronoun</td>
</tr>
<tr>
<td>13.IDF-PL: plural indefinite pronoun</td>
</tr>
<tr>
<td>14.INF: infinitive marker</td>
</tr>
<tr>
<td>15.INT: interjection</td>
</tr>
<tr>
<td>16.LET: letter</td>
</tr>
<tr>
<td>17.LET-PL: plural letter</td>
</tr>
<tr>
<td>18.LOC: locative adverb</td>
</tr>
<tr>
<td>19.MOD: modal auxiliary</td>
</tr>
<tr>
<td>20.N: noun</td>
</tr>
<tr>
<td>22.N-PR: proper noun</td>
</tr>
<tr>
<td>23.NEG: negation</td>
</tr>
<tr>
<td>24.NEO: neologism</td>
</tr>
<tr>
<td>25.NUM: number</td>
</tr>
</tbody>
</table>
3 Evaluation

3.1 Four taggers

Comparing the errors of the different taggers will allow us to see what kinds of assumptions about word category assignment lead to error. As of four taggers have been chosen in this investigation, each one of the taggers stands for a kind of tagging algorithm: the CLAN MOR tagger as rule based combined with stochastic model, Version 1.14 of Brill as transformational based model, ACOPOST trigram tagger as a simple statistical based model, and Stanford tagger as a max entropy model. We will briefly introduce these four taggers in this session.

3.1.1 CLAN MOR tagger

The CLAN tagger has two components: MOR and POST. The MOR component lemmatized words: it breaks words down into their morphological component, and tags each morpheme for all its possible parts of speech. The POST component disambiguates MOR’s output, choosing the most probable tag. For example, in the sentence ‘A can/noun can/auxiliary can/verb a can/noun’, the possible tags for ‘can’ are noun, auxiliary and verb. The MOR component saves three tags for each ‘can’. The POST component establishes that
the first ‘can’ in the sentence is a noun, the second ‘can’ is an auxiliary, and the last ‘can’ is a verb. MOR component uses a lexicon and morphological rules to provide all the possible parts of speech and morphological analyses for each word. POST component chooses one single interpretation for each word from the options provided by MOR. The combination of the two will henceforth be referred to as MOR tagger. The version of CLAN MOR tagger used in this project is V 28-Dec-2014 11:00.

Specifically, POST uses information from the context surrounding the tag to make disambiguation decisions. The process through which the POST component gains a set of rules for making these disambiguation decisions is called ‘training’. Files containing sentences with unambiguous and correct parts of speech tags for each word referred to as ‘training corpus’. Transcribed utterances in the CHILDES corpora are potential training corpus for MOR tagger.

After the training session, the POST component gains statistical rules. These rules describe which pairs of tags have been detected in the training corpus, and how frequently they have appeared. By considering the frequencies of all consecutive pairs of tags in a sentence, POST determines the most likely tag for each word.
MOR tagger was developed through training on Eve’s corpus and additional adult speech. The exact size of the training set is not clear, but it was probably between 75,000-100,000 words. MOR has 31 basic tags, but by combining part-of-speech tags and morphosyntactic labels, it is able to create considerably more tags, although there is no documentation for every tag.

3.1.2 ACOPOST tri-gram tagger

ACOPOST (formerly known as ICOPOST) is a collection of four kinds of taggers: a Maximum Entropy Tagger (MET), a Trigram Tagger (T3), an Error-driven Transformation-based Tagger (TBT), and an Example-based Tagger (ET). The ACOPOST toolkit (Schroder 2002) is freely available under the GNU public license from the author’s home page.

Each tagger in the ACOPOST toolkit implements one kind of part-of-speech tagging algorithms. MET is built on the framework of Ratnaparkhi (1997b); TBT is built on the work of transformational rules from Brill (1993); ET is built on the work of Daelemans et al. (1996), and T3 is built on the work Brants (2000). In this project, we choose to use T3 because the implementation is closely follows the standard Hidden Markov Models (HMM)
approach.

Hidden Markov Model is a stochastic finite state automaton with probabilities for the transitions between states and possible output tokens. In a simple Markov Model, every state is “known” to the observer, and the probability of each transition is clear. For a hidden Markov Model, the probability of each transition is “unknown” to the observer, but the possibility of an output token of each transition is “known” to the observer. N-gram Tagger picks the most statistically likely tag for each word in one sentence, i.e. actually calculate the probability of a given tag sequence using context. The sequence of tags, which has the highest score, is the final tagging result.

For example, in the sentence of ‘A/determiner can/noun can/auxiliary can/verb a/determiner can/noun’, the Trigram Tagger counts two kinds of possibilities. The one is the possibility of a certain tag to a word: The word “can” might be a noun, an auxiliary, or a verb in this sentence. Then the tagger counts the possibility of “can” being a noun (50% in this example), an auxiliary (25%), and a verb (25%). The other possibility is about a tag appears after two consecutive tags, such as the possibilities of an auxiliary appears after two consecutive tags ‘determiner-noun’, a verb appears after ‘noun-auxiliary’, a determiner
appears after ‘auxiliary-verb’, and a noun appears after ‘verb-determiner’. The Trigram Tagger calculates these possibilities based a second order Hidden Markov Model to assign a word its part of speech tag. During the training process, the Trigram Tagger in ACOPOST toolkit uses relative frequencies supervised learning. It uses deleted interpolation for smoothing, and the Viterbi algorithm for decoding.

3.1.3 Brill tagger

Eric Bill invented this tagger and described it in his 1995 PhD thesis. It is a typical transformational-based tagger. It shares features of both rule based and stochastic tagging architectures. In the training process, Brill tagger automatically learns words and rules from previously tagged training data. These rules are used to disambiguate among possible tags. In the tagging process, Brill tagger firstly assigns each known word to the most frequent tag. All words that appear in the training data are “known words”. For example, “can” is a noun in “This is a can of coffee”; and a modal auxiliary in “I can do it.” “Can” is more often a modal auxiliary than a noun; Brill tagger thus assigns it a tag of modal auxiliary.

There were words that did not appear in training data. These words are called “unknown words” in the tagging process. Brill tagger assigns by default the most common tag to
unknown words. For instance, “vig” is a new word that was made up in the following conversation:

“I think Tom is one very ignorant guy.”

“Hmm, he’s a big vig.”

“Vig” is an unknown word to taggers. Brill tagger would assign it a noun tag because noun is the most frequent tag in most contexts. After the first round of assigning tags, Brill tagger applies learned rules to the data; and changes the incorrect tags. Brill tagger does this “apply rules and change” process repeatedly until it reaches a high accuracy, or until there are no more rules that can be applied.

In the same example, in the sentence of ‘A/determiner can/noun can/auxiliary can/verb a/determiner can/noun’, Brill tagger assigns all ‘can’ an auxiliary tag, and changes ‘auxiliary’ to ‘noun’, or to ‘verb’ based on the rules that learned from training process.

 Compared to many other stochastic taggers, which need tens of thousands of lines of statistical rules to capture contextual information, Brill tagger is a simple part of speech tagger. If the tagger were trained on a different corpus, a different set of tagging rules would be learned automatically. Therefore, Brill tagger is portable and readily transferable to a
different tag set, text genres or foreign languages.

3.1.4. Stanford tagger

The Stanford parser is based on discriminative models. The tagger learns a log-linear conditional probability model from tagged text, using a maximum entropy method. It picks the “best” tag using information from context and word features (Dan Klein and Christopher D. Manning, 2003). The reported accuracy of Stanford tagger is 97.24% on the Penn Treebank Wall Street Journal (Toutanova, 2003). The original English tagger uses the Penn Treebank tag set. On its official website, there are tagger models in Arabic, Chinese, English, French and German. However, Stanford tagger can be trained in any language if the training data for the language is annotated with part of speech tags. In this project, we used the 2010 version of Stanford parser. We also used LARC tag set instead of Stanford Treebank tag set.

3.2 Methodology and Tagging accuracy

The LARC tagged data was used as the “gold standard” in all tags in the training process to keep the consistency of our experiment result. We tested taggers with three different sizes of training data in this project.

<table>
<thead>
<tr>
<th>Round</th>
<th>Utterance Number</th>
<th>Word Counts</th>
<th>Utterance Number</th>
<th>Word Counts</th>
<th>Utterance Number</th>
<th>Word Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2885</td>
<td>9315</td>
<td>5,770</td>
<td>18,630</td>
<td>11,540</td>
<td>37,960</td>
</tr>
<tr>
<td>B</td>
<td>2885</td>
<td>16158</td>
<td>5,770</td>
<td>32,316</td>
<td>11,540</td>
<td>64,852</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
As shown in Table 3, in the first round (Round A), we trained and tested four taggers using 2,885 utterances of children’s speech and adult’s speech, respectively. Both children’s speech and adult’s speech used in this round of experiment are from Valian children’s data. In the second round (Round B), we doubled the size of training and testing data to 5,770 utterances. Both children’s and adult’s speech in the training and testing process are from Valian data as well as in round A. This means we used all children’s data, and half of the adult’s data from Valian corpus. In the third round (Round C), we doubled the utterance number of gold standard data again. We trained and tested four taggers with 11,540 utterances of children’s speech and the same mount of adult’s speech. For this round, the children’s data are a half from Valian corpus, and a half from Eve corpus at parallel mean length of utterance. The adult’s data are all from Valian corpus.

10-fold cross validation was used to calculate the accuracy of a tagger during all three rounds of evaluation processes. We divided the gold standard data into ten equal folds using stratified sampling by the sentence length. Then we used with nine folds of the data for training. In the training process, the tagger “learned” the language model from the training data set based on statistical possibility. The remaining one fold was for testing. In the testing
process, the tagger assigned tags to the testing data set (a version of data without tags) with
the learned language model. After this, we yielded a tagged version of the testing data set.
We evaluated the tagged version of the testing data with its “gold standard” version, and get
accuracy for one fold of the data set. After ten times of training and testing, the whole data
set was tested. We then calculated the average of ten accuracies as the final accuracy of the
tagger. The tagging accuracy therefore describes how well the tagger performs.

To set up a benchmark case for tagging accuracy, a uni-gram tagger was trained and
tested with the same data set in this project. Uni-gram tagger was introduced by Eugene
Charniak in *Statistical Techniques for Nature Language Parsing* (AI Magazine, 1997). The
construction of uni-gram tagger is quit straightforward. Tagger learned each word token and
found its most frequently appeared tag from the training data. For example, if training and
testing the uni-gram tagger with the same sentence ‘a/determiner can/noun can/auxiliary
can/verb a/determiner can/noun’, the uni-gram tagger would learn to give “a” the tag of
“determiner”, “can” the tag of “noun”. Because “determiner” is the most frequent tag to the
word “a”, and “noun” is the most frequent tag to the word “can” in the training sentence. So
the sentence would be tagged as “a/determiner can/noun can/noun can/noun a/determiner
can/noun” by the uni-gram tagger in this case, disregarding the possibility that “can” would be tagged as “auxiliary” or “verb”. In this project, we called a simple uni-gram tagger function (UnigramTagger) from tag class (nltk.tags) of NLTK toolkit. The results are listed in Table 4.

| Table 3: Unigram, Brill, Trigram, and Stanford tagger accuracies (%) |
|---|---|---|---|
|                      | Unigram | Brill | Trigram | Stanford |
| Child - Round A      | 90.26    | 91.17 | 94.08    | 93.45    |
| Child - Round B      | 92.52    | 93.67 | 95.81    | 95.27    |
| Child - Round C      | 89.54    | 90.44 | 92.51    | 90.63    |
| Adult - Round A      | 88.57    | 89.97 | 93.55    | 94.11    |
| Adult - Round B      | 90.42    | 91.93 | 94.27    | 95.60    |
| Adult - Round C      | 90.87    | 92.29 | 95.44    | 95.64    |

For the overall tagging accuracy in three rounds, tri-gram tagger performed better with child’s speech, and Stanford tagger gained a better result on adult’s speech. The uni-gram tagging accuracy was 90.26% in round A of child’s speech. As the mount of training data doubled, the accuracy reached 92.52% in round B. The raising of accuracy from round A to round B was observed on all three taggers in both child’s speech and adult’s speech. On child’s speech, tri-gram tagging accuracy grew from 94.08% to 95.81%, increasing 2.5%. Stanford tagging accuracy grew from 93.45% to 95.27%, increasing 1.73%. And, the Brill tagger tagging accuracy went from 91.17% to 93.67%, increasing 1.82%, which was the largest growth among all three taggers. On adult’s speech, the uni-gram tagging accuracy
grew from 88.57% to 90.42%, increasing 1.85%. The Brill tagging accuracy grew from 89.97% to 91.93%, increasing 1.96%. The tri-gram tagging accuracy went from 93.55% to 94.27%, increasing 0.72%. And the Stanford tagging accuracy went from 94.11% to 95.60%, increasing 1.49%.

Generally speaking, the tagging accuracy will get better if increasing the mount of training data; however, all the taggers showed a drop of tagging accuracy in round C, which has the largest training data set among three rounds. The uni-gram tagging accuracy turned out to be 89.54% on child’s speech in round C. Tri-gram tagging accuracy changed from 95.81% to 92.51%, reducing 3.30%. Brill tagging accuracy dropped from 93.67% to 90.44%, reducing 3.23%. The tagging accuracy of Stanford tagger dropped from 95.27% to 90.63%, creating a 4.64% gap. On adult’s speech, tagging accuracy was slightly better in Round C. The uni-gram tagging accuracy was 90.87% on adult’s speech in round C, increasing 0.45%. Tri-gram tagging accuracy changed from 94.27% to 95.44%, increasing 1.17%. Brill tagging accuracy changed from 91.9% to 92.3%, increasing 0.36%. And the tagging accuracy of Stanford tagger changed from 95.60% to 95.64%, increasing 0.04%.

3.3 Problematic tagging cases
To look closely at the tagging results, we made following confusion matrices from the tagging results of the tri-gram tagger, Brill tagger and Stanford tagger in round B. In round B, we trained three taggers with all child’s speech data and the same mount of adult’s speech data from Valian corpus, and every tagger gets a overall best result among three rounds.

For each tagger, the child’s testing result is listed on the left hand side, and the adult’s testing result is listed on the right hand side. Column and row numbers represent corresponding taggers as listed in Table 2, List of LARC Tags. Each row is one part of speech category that classified as different tags into each column. From blue to red, the color codes the number of words that was classified into different word category. This means the diagonal represents the correctly tagged word counts. Every color dot appears above or under the diagonal is an incorrectly tagged result.

Figure 1: Confusion Matrix of ACOPOST Trigram Tagging Result in Round B
To better illustrate the testing result of three taggers, we listed the precision rate of each tagger in one figure on the left side of Figure 4. The recall rate of each tagger is listed on the right side in Figure 4.
By a carefully looking at the transcript of the testing data, we spot some problematic tagging cases. “N” and “ADJ” are among these cases. “N” in the sentence of “Wool sweaters make my neck itch” was tagged as “ADJ”. “N-PL” in the phrase of “Chinese celebrate the
festival” was tagged as “ADJ”.

Particles (PTL), prepositions (PRE) and adverbs (ADV) are very difficult to distinguish. The first criterion to check is whether or not the word in question, which we will call a “little word” for convenience, is followed by a noun phrase. If the “little word” is followed by a noun phrase that may be its object or the object of the verb plus “little word” combination, the relevant distinction is between preposition and particle. Prepositions take noun phrase objects while particles do not. Rather, a verb-particle construction that is transitive takes a noun phrase object. There is a test that to move the little word to the right of the noun phrase. If both orders are grammatically have the similar meaning of the sentence, than the word should be tagged as particle. One can rely on this test to illuminate this distinction. But it is very hard for automatic taggers to do this task and assign the correct tag.

QNT (quantifier) and N (noun) is another confusing case. A feature particular to quantifiers is their ability to enter into ambiguities of scope. Sentences in which the object and subject are quantified have two interpretations deriving from the relative scope of the two quantified noun phrases. For example, every/QNT cat loves some/QNT dog can mean that there is a dog that every cat loves or that for each cat there is a dog that this cat loves. This is
different from nouns that express quantity such as “a bunch”, in the sentence of “a bunch/N of cats love some/QNT dog”.

Due to the limit of space, we only listed some of the confusing tagging cases. Above-mentioned cases are typical ones that can be found via a detailed examination of the results from confusion matrix, and a hand check with the transcript of the testing data.
4 Qualitative Error Analysis

4.1 CLAN MOR tagging difference on Valian and Eve corpus

The construction of CLAN MOR tagger is different from the other three taggers we use in this project. CLAN MOR tagger morphologically analyzes each words in the sentence before assign it a part of speech tag. The LARC tagset does not include the morphological token, so it could not parse words into its morphemes. This makes it difficult to run the CLAN MOR tagger using LARC tagset. As described in the corpus construction section, we manually checked the tagging result of CLAN MOR tagger to make our “gold standard”. Every error the CLAN MOR tagger made when dealing with Valian corpus was recorded on file. So we did not run CLAN MOR tagger again, but to calculate the accuracy based on former correction files. The CLAN MOR manual includes a detailed error analysis of CLAN MOR tagger’s performance on Eve corpus. It’s accuracy on Valian corpus and Eve corpus are listed below.

<table>
<thead>
<tr>
<th>Table 4: CLAN MOR Tagging Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valian - Child</td>
</tr>
<tr>
<td>Valian - Adult</td>
</tr>
</tbody>
</table>

The CLAN MOR tagger was trained on 20 files from Eve corpus. When it tags on Eve corpus again, the training and testing data set are derived from the same data source. The
high similarity may over-fit the testing data, thus CLAN MOR tagger gains a higher accuracy in Eve corpus than in Valian corpus.

4.2 Tagging accuracy and the size of training data

With the increasing mount of training data, taggers supposed to learn more rules from the training process and thus have a better performance on testing result. The tagging result from Round A to Round B shows a strong evident for this presumption. In contrast, the testing result of round C is decreased from round B. There are a few possible explanations for this reduction, such as different mean length of utterance and word-tag distribution in Valian and Eve corpus. To verify the possible reasons for the changes on tagging accuracy, we have done several analyses as described below.

The Valian corpus and Eve corpus do have different words per tag distribution. We examined the words per tag distributions of Valian data used in testing round B. For illustration purpose, the number of word count was logarithmic transformed.
From figure 6, we can see seven POS tags didn’t appear in Valian corpus. They were “FIL: filler”, “GER-PL: plural gerund”, “IDF-PL: indefinite pronoun”, “LET: letter”, “LET-PL: plural letter”, “LOC: locative adverb”, “NEO: neologism”, “NUM-PL: plural number”, “PHO: phonological word”, “SIN: sing word”, “TEM: temporal adverb”, and “TEST: test word’. The reason of missing these part of speech tags could be that the corpus doesn’t have them in the original script, such as LOC and TEM are tagged as general adverb in Valian corpus. There is another possibility that some of the features of the tag disappeared in the process of applying correction. To test this assumption, a more detailed screening or manual checking is needed for further research.

The mean length of utterance may play a role in the decreasing of tagging accuracy while the training data set is increasing. The mean length of utterance (MLU) distribution is also different in Valian corpus and Eve corpus.

| Table 5: Mean Length of Utterance in training data from Valian corpus |
|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
|                 | 1a  | 1b  | 2a  | 2b  | 3a  | 3b  | 4a  | 4b  | 4c  | 5a  | 6a  | 6b  | 7a  | 8a  | 8b  | 15a | 15b |
| CHI             | 1.93| 1.74| 2.09| 1.91| 1.81| 1.90| 2.07| 2.14| 2.71| 2.12| 2.28| 2.69| 2.31| 2.59| 2.94| 3.92| 3.79 |
| ADU             | 5.66| 5.51| 4.96| 4.72| 4.28| 4.91| 6.95| 6.72| 6.26| 4.88| 5.69| 5.53| 4.47| 4.82| 5.10| 4.63| 4.54 |

In table 6, we can see the mean length of utterance (MLU) in child’s training data from Valian corpus was continuously increasing from file “1a” to “8b”. Whereas in the file “15a”
and “15b”, MLU jumps to 3.92 and 3.79, creating a gap around 0.85 - 0.98.

<p>| Table 6. Mean length of utterance in child's data from Eve corpus |
|--------------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|</p>
<table>
<thead>
<tr>
<th>CHI</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.45</td>
<td>1.51</td>
<td>1.85</td>
<td>1.65</td>
<td>1.90</td>
<td>2.23</td>
<td>2.11</td>
<td>2.42</td>
<td>2.79</td>
<td>2.78</td>
<td>2.77</td>
<td>3.13</td>
</tr>
</tbody>
</table>

In table 7, the mean length of utterance in child’s data from Eve corpus also showed a change between each file, but these changes ranged from 0.06 to 0.36. The assumption that different MLU affects tagging results might be challenged by the result from round A to round B. For round A and round B both using the same training data from Valian corpus, taggers performed better in round B than round A. However, the training data set used in round A was quite small. The influence factor could be the size of training data, or the utterance length of the training data. It is largely unknown which factor contributes more to this problem. One of the further researches can be training and testing a small mount of data with continuously increasing mean length of utterance.
5 Conclusion

In this project we evaluated four mainstream taggers in three rounds, controlling the testing data set. These taggers are CLAN MOR tagger, ACOPOST trigram tagger, Stanford parser, and Ver 1.14 of Brill tagger. The child-adult’s dialogues of 18 files from Valian corpora and a counter part of Eve corpora have been manually labeled, and rewrote with LARC tagset. They served as gold standard corpora in the training and testing process.

We carefully analyzed the ground constriction of each tagger. By analyzing what kinds of assumptions the tagger made about category assignment lead to failing, we identify several problematic cases of tagging, such as “ADJ” vs “N”; “PTL” vs “PRE” vs “ADV”; and “QNT” and “N”. A further discussion of this part could be founded with the listing results of tagging accuracy of each tag and the original transcript of the child-adult’s dialogues.

Our testing results showed that the tri-gram tagger had a better performance on child’s speech data, and the feature rich Stanford tagger has a better performance on adult’s speech data. By comparing the average error rate of each tagger, we found that the size of training data set largely affected the tagging result. Generally speaking, the more data were used in training process, the higher accuracy could be reached. However, this statement has a
prerequisite when dealing with child’s speech. The child’s speech is somewhat arbitrary than adult’s speech. To gain a reasonable tagging result, taggers would be better trained and tested in a very similar data sets, which means a similar length of utterance, or a similar language model.

In addition, the length of utterance also played a role to affect tagging accuracy. But it is still not clear how it influenced the tagging result. Our immediate plans include continued study on different size of training data with similar length of utterance. We also plans to spare a part of the data as a constant testing data, in order to set up a reference for each tagger under different training environment. As future work, we plan to analyze the nature of tagging errors, and develop a more accurate tagger or set of taggers. This accurate tagger will allow full utilization the CHILDES database, and thus place language acquisition analyses on a firmer footing.
Reference


