Morphosyntactic Analysis of the CHILDES and TalkBank Corpora

Brian MacWhinney
Carnegie Mellon University, Psychology
5000 Forbes Ave. Pittsburgh PA 15213 USA
E-mail: macw@cmu.edu

Abstract

This paper describes the construction and usage of the MOR and GRASP programs for part of speech tagging and syntactic dependency analysis of the corpora in the CHILDES and TalkBank databases. We have written MOR grammars for 11 languages and GRASP analyses for three. For English data, the MOR tagger reaches 98% accuracy on adult corpora and 97% accuracy on child language corpora. The paper discusses the construction of MOR lexicons with an emphasis on compounds and special conversational forms. The shape of rules for controlling allomorphy and morpheme concatenation are discussed. The analysis of bilingual corpora is illustrated in the context of the Cantonese-English bilingual corpora. Methods for preparing data for MOR analysis and for developing MOR grammars are discussed. We believe that recent computational work using this system is leading to significant advances in child language acquisition theory and theories of grammar identification more generally.

Keywords: morphosyntax, part of speech tagging, spoken language corpora, child language, morphology, dependency grammar**:**

1. Introduction

This paper describes the construction and usage of computational systems for the morphosyntactic analysis of the spoken language data in the CHILDES (http://childes.psy.cmu.edu) and TalkBank (http://talkbank.org) databases. CHILDES contains 60 million words of child-adult conversation across 26 languages; TalkBank includes 63 million words of adult-adult conversation with the bulk in English. The data collected in these corpora come from 128 separate projects conducted over the last 40 years. All of the data are in a transcription format called CHAT that can be automatically converted to the TalkBank XML format (http://talkbank.org/software/xsddoc) for validation and analysis through other tools. Newer corpora have been transcribed directly into the TalkBank format, but older corpora were reformatted to match the current format. TalkBank includes corpora in eight diverse areas including aphasia, conversation analysis, gesture, bililngualism, classroom discourse, legal arguments, dementia, and second language acquisition. We think of CHILDES as the subset of the larger TalkBank system that focuses specifically on child language. Together, the various TalkBank corpora constitute the largest available corpus of consistently transcribed spoken language materials. Nearly all of the transcripts in TalkBank are linked on the utterance level to either audio or video. However, for the CHILDES segment, only about 25% of the transcripts are linked to media.

2. The MOR Program

From the beginning of this project in 1984, researchers have been interested in conducting morphosyntactic analyses of these resources. Initially, we hoped to adapt off-the-shelf morphological taggers for this process. Because TalkBank data derive from many different languages, we would need to have methods that could be easily adapted to each target language. We found that existing taggers had a variety of limitations. Many were not open-source, making further development difficult. The lexicons used by the taggers focused primarily on written, rather than spoken speech, seldom including methods for tagging interjections, onomatopoeia, babbling, code mixing, and many other forms in natural spoken language. The FSM finite-state morphology framework (Beasley & Karttunen, 2003) seemed promising, but few of the taggers developed in that framework were available for further development or sharing. Moreover, the construction of FSM/XFST taggers requires a level of computational skill not available to some linguists. As Wintner (2007) notes, the greater memory capacity and speed of modern computer makes it feasible to approach morphological analysis through allomorph generation (the approach taken in MOR), rather than being limited to recognition (the approach taken in XFST). This also provides better control of overgeneration and debugging information. Given these various considerations, we decided to build a system for part-of-speech (POS) tagging through generation that could operate across a wide variety of languages. Crucially, we wanted to have a system that would provide the non-programmer with easy ways to add new lexical forms to the system.

This program, called MOR, was originally designed by Roland Hausser and later extended by Mitzi Morris and Leonid Spektor. MOR grammars have now been built for English (Brian MacWhinney), Spanish (Brian MacWhinney), Japanese (Susanne Miyata), Mandarin (Brian MacWhinney and Twila Tardif), Cantonese (Sampo Law, Anthony Kong, and Brian MacWhinney), French (Christophe Parisse), Italian (Brian MacWhinney), German (Heike Behrens and Brian MacWhinney), Dutch (Steven Gillis), Danish (Brian MacWhinney), and Hebrew (Shuly Wintner, Aviad Albert, and Bracha Nir). Taggers are under construction for Afrikaans, Swedish, Norwegian, Portuguese, and Farsi. All of these grammars are downloadable from http://childes.psy.cmu.edu/morgrams, and the MOR program that runs them is included in the CLAN program that can be downloaded from http://childes.psy.cmu.edu/. The manuals for the CHAT

and CLAN programs were last published in paper form in 2000 (MacWhinney, 2000) and current versions of the documentation can be downloaded from http://childes.psy.cmu.edu/manuals.

3. Word Forms

Because the major target of MOR grammars is spoken language, it is important to include consistent representations for informal forms such as interjections, communicators, and dialect forms. We also try to keep the POS tags consistent across languages, but there are many types unique to particular languages, such as final sentence particles in Chinese or English borrowings in Welsh. There is a strong emphasis in MOR lexicons on the analysis of compound forms. For English, there are 87 separate files for each of the 87 part-of-speech types. Of these, 27 involve different types of compound formation. For example, we distinguish compounds like "birdbath" formed from two nouns from compounds like "wood+working" composed of a noun followed by a participle. Within the interjections, we distinguish monomorphemic interjections such as "amen" from compound interjections such as "good+morning". We have also begun work that treats word combinations such as "so that" or "in order to" as single conjunctions written in the form "so_that" or "in_order_to". We have found that consistent treatment of these forms as combinations leads to corresponding improvements in syntactic analysis.

As we refine our analysis of lexical items, we also reconcile changes with the database by running global replacement sequences and then using the MOR program to check to see that all words are recognized. This is done by the command "mor +xl *.cha" which runs the MOR grammar across a collection of files and creates an entry for an words that are not being analyzed. This is an easy process, because MOR takes only seconds to run across the files in a given corpus. However, when words are not recognized, the process of either changing the lexicon or modifying the forms in the corpus can take much more work, particularly for corpora that have not yet been analyzed by MOR.

4. Lexical Entries

The shape of lexical entries is quite simple. Words are entered into text files one word on each line in alphabetical order. The surface form comes first on the line, followed by the scat or syntactic category, some possible morphological features, and a possible English translation. Here are some examples from two different POS files in Spanish:

```
este {[scat det:dem][inflect yes]} =this=
abotona {[scat scat v]} =button=
```

The most complex entries are usually for irregular verbs, as in this example:

```
dé {[scat vimp][allo irr]} "da-3S&IMP" =give=
```

In these forms the material in quotes is used to provide a morphological analysis that sidesteps the basic analysis provided for the bulk of the lexicon. Only a few highly irregular forms are treated in this way.

5. Components of Words

For each language, there is an emphasis on the extraction and representation of the full linguistic form of each morphologicallly complex word, including affixal, inflectional, and clitic structures. However, true transparent combination is distinguished from fusional morphology by using the dash (–) mark for the former and the ampersand (&) mark for the latter. For example, here is a morphologically tagged sentence from the Hebrew Berman-Longitudinal database with the mother asking, "What he do?"

```
*MOT:   ma huʔ ʕoṣē ?
%mor:   que|ma=what
        pro:person|huʔ&gen:masc&num:sg=he
        part|ʕaṣā&root:ʕṣy&ptn:qal&gen:masc&nu
        m:sg=do
```

Here is an example sentence from the Spanish corpus:

```
*MOT:   vamos a dormir .
%mor:   vpres|i-1P&PRES=go
        prep|a=to vinf|dormi-INF=sleep .
```

The possible components of words occur in this order:

prefixes, marked at the end with #
the syntactic category of the stem
a turnstile or bar character (|)
the citation form of the stem
affixes, marked by – or & at the beginning

In addition, the clitic marker ^ can be used to separate the two components of cliticized combinations, as in mod|do~neg|not for "don't". Compounds are represented as illustrated by this form for "blackboard".

```
n|+adj|black+n|board
```

6. Allomorphy Rules

The construction of these morphological analyses depends first on the generation of a runtime lexicon compiled through the operation of rules of allomorphy (arules), as they operate on the items listed in the lexicon. The full set of generated allomorphs is stored in a trie structure (Fredkin, 1960). Here is an example of the allomorphy rule for final consonant doubling in English spelling:

```
LEX-ENTRY:
LEXSURF = $O$V$C
LEXCAT = [scat v], ![tense OR past perf], ![gem no]
ALLO:
ALLOSURF = $O$V$C$C
ALLOCAT = LEXCAT, ADD [allo vHb]
ALLO:
ALLOSURF = LEXSURF
ALLOCAT = LEXCAT, ADD [allo vHa]
```

Here, the string OV$C is composed of variable declarations that characterize the final VC pattern in verbs like "bat". Overgeneration of the rule to produce forms such as "putting" from "put" is blocked by inclusion of the feature ![gem no] in the rule and [gem no] in the lexical entry for "put". The first allo generated by the rule is "batt" which will produce words like "batting" or "batter" and the second is "bat" which will produce "bats" or "bat". The actual use of this lock-and-key allomorphy pattern matching mechanism is

controlled by the crules, discussed below.

The application of arules is strictly ordered. If a lexical entry matches a given rule, that rule applies and later rules can no longer apply to that form. This is equivalent to the "bleeding" pattern of generative phonology. However, there is no "feeding" relation in MOR, because the output of one rule cannot serve as the input to another. Given this, it is important to order arules so that the most specific rules for irregular forms comes first and general patterns come last. Apart from attention to rule ordering, it is important to control the lock-and-key matching system for feature-value pairs through careful documentation and control of the [allo] features and other grammatical feature-value pairs such as [gen]. A complete list of allomorphy and other grammatical types for English is given in the file engcats.cdc in the /docs folder in the MOR grammar.

7. Concatenation Rules

MOR operates on words in a corpus one at a time. Each word is defined as a series of characters delineated by surrounding spaces. For each word being analyzed, MOR goes through the word letter by letter, attempting to match the current input string to one of the allomorphs in the runtime lexicon. This matching process is governed by the second set of MOR rules – the concatenation rules or crules. These rules are applied using feeding relations and no bleeding relations. This means that all rules that could apply to a given input will fire, sometimes producing multiple possible threads. However, a thread will only be outputted if it tags the complete word. Therefore, many candidate threads will fail along the way. At the beginning of the word, matches are determined by the START rules that only require that a morpheme match the syntactic category or scat of the rule. After the first rule match creates a candidate for the first few letters, MOR continues to take in letters looking for another morpheme match. Once a new morpheme fires, there can be lock-and-key process in which the STARTCAT and the NEXTCAT must match in terms of their allo features. This is the MATCHCAT process. Here we will look at some crules that illustrate some of these processes:

```
RULENAME: bare-start
CTYPE: START
 if
  NEXTCAT = [scat OR co co:voc conj]
 then
  RESULTCAT = NEXTCAT
  RULEPACKAGE = {}
```

This bare-start rule takes words that receive no morphological analysis and sends them directly to the output. The actual list of forms in the fourth line of this rule is much longer.

The next sample rule is used to create gerunds from verbs:

```
RULENAME: n:v-deriv
CTYPE: -
 if
  STARTCAT = [scat OR v v:cop], ![bare yes]
  NEXTCAT = [scat n:gerund]
  MATCHCAT [allo]
 then
  RESULTCAT = NEXTCAT, STARTCAT [comp],
      DEL [allo], ADD [allo n0]
  RULEPACKAGE = {n-pl, n-cl}
```

Here, the CTYPE line shows that the suffix –ing for the gerund is being attached as a combinatorial affix. The two rules that feed into this rule both apply to words that are [scat v], so restating this restriction in the STARTCAT line is a bit redundant, but good for clarity. The feature of the NEXTCAT comes from the entry for the suffix, which is stored in the 0affix.cut file in the directory of lexicon files. That entry also includes a list of the six verb stem allomorph types with which the suffix can combine. The MATCHCAT [allo] process makes sure this match is correct. The RESULTCAT rule makes sure that the output includes information that the form is a gerund and that it includes the compound structure of the input verb for gerunds such as "baby+sitting". That rule also removes features that were only important for the MATCHCAT process. Finally, the RULEPACKAGE line sends this candidate form on to further analysis by other rules. This would allow analysis of forms such as cliticized "singing's" as in "singing's not my forté".

Once all the letters of the input have been recognized, a form goes to the endrules (CTYPE: END) that write all forms to the output, unless they violate some specified condition. Thus, these rules can serve as a final filter to block overgeneration. There are, therefore, several ways to avoid overgeneration in MOR:

1. The allomorphy rules are strictly ordered so that matches of earlier lexically-specific or limited rules bleed out context for later more general rules.
2. Overapplication is also controlled through the use of MATCHCAT checking between stems and affixes. Most of these features are for general categories, such as gender or allomorphy type, but some, such as [prefix no] are used to block specific disallowed forms.
3. Items that cannot appear without inflections are given the feature "bare" which is removed during affix attachment. During the endrule of the crules processing, forms that still contain the "bare" feature are blocked.
4. Iregulars can be given full listings.

The final output of MOR can also include certain grammatical features that are inserted as non-concatenative morphemes on the stem. For example, in Spanish, the gender of nouns is always printed out as either &FEM or &MASC so that further analysis can use this information. The output.cut file in MOR controls the prinint in the output of these non-concatenative features

8. Grammar Development

Once a MOR grammar exists for a language, application of that grammar to a new corpus involves basic lexical work and error checking. Because the English MOR grammar is stable and robust, the work of analyzing a new corpus seldom involves changes to the rules in the ar.cut or cr.cut files. However, a new English corpus is

still likely to need extensive lexical clean up before it is fully recognized by MOR. The unrecognized words can be identified quickly by running this command:

mor +xl *.cha

This command will go through a collection of files and output a single file "mini lexicon" of unrecognized words. The output is given the name of the first file in the collection. After this command finishes, the user must open up the file to see all the words not recognized by MOR. There are several typical reasons for a word not being recognized:

1. It is misspelled.
2. The form is a nonword that should be preceded by an ampersand (&) to block look up through MOR.
3. The word should have been transcribed with a special form marker, as in bobo@o or bo^bo@o for onomatopoeia.
4. The word was transcribed in "eye-dialect" to represent phonological reductions. To maintain this coding, use forms such as gonna [: going to].
5. Proper nouns need to begin with capitals.
6. The stem needs to be entered into the lexicon.

For languages that do not yet have a MOR grammar, one must be created. This can involve several weeks of intensive work. For languages with rich morphology, it is important to be guided by a systematic textbook of inflectional and derivational patterns. For example, for Spanish we relied on the Berlitz verb book (Berlitz, 2005). Using these analyses, the first step is to determine the basic allomorphy types of the language and to assign them consistent abbreviations. In some cases, these patterns can be based on phonological processes. This works in languages like English or Danish. However, in languages with more complex paradigms, such as Spanish or Hebrew, it is better to create allomorphy types based on the formal segments of the nominal or verbal paradigm. For example, Spanish verbs can be given stems that specifically target the preterite, the subjunctive, and so on. For some verbs, multiple preterite stems must be generated, whereas for others a single preterite stem is sufficient.

9. Disambiguation

The output of the MOR program is a new tier or line called the %mor line in which tags stand in one-to-one correspondence with words on the main transcript line, excluding non-words and repetitions. However, this format is not yet disambiguated. Words can receive as many as 6 different analyses, all concatenated with the caret (^) symbol. To achieve disambiguation of such combinations, we use the POST and POSTTRAIN programs written by Christophe Parisse. These programs use a gold standard training corpus to train a statistical disambiguator.

The training corpus includes a %trn line that represents the target values to which POST should adapt. This line is created by hand (with occasional bootstrapping) and must be modified by hand. In terms of development work, it is important to keep the tags and features of the %trn line in accord with those produced by MOR. As rules and features in MOR change, the forms in the %trn line must be edited so that there is always a match. Otherwise, errors will be reported during the running of POSTTRAIN.

The combination of MOR and POST for the English CHILDES database yields tagging that is accurate at a level between 95 and 98%. We have also applied English MOR extensively to the adult transcripts in the AphasiaBank segment of TalkBank and observed an accuracy level of 98% for the normal control participants, although the level for the aphasics is unsurprisingly lower. For MOR and POST, we only report accuracy scores, because precision is always at 100% by stipulation. This is because MOR recognizes all words, given our procedure.

For English, many of the remaining errors include problems discriminating nouns and verbs, particularly in one- or two-word utterances produced by young children. For highly inflected languages like Hebrew or Spanish, there are few errors, but occasional problems with ambiguity in the declensional or conjugational paradigms.

10. Tagging Bilingual Corpora

It is also possible to systematically tag bilingual or multilingual corpora, as long as MOR grammars exist for each language included. Currently, the best example of this form of analysis can be found in the Cantonese-English child language corpus in CHILDES contributed by Virginia Yip and Stephen Matthews of the Chinese University of Hong Kong. In this corpus, each file includes this header line:

@Languages: eng, yue

In this case, English is taken as the default language for the transcript, i.e. the language of the majority of the utterances. In that case, each Cantonese utterance is marked at the beginning with [- yue]. If a single word or a few single words in an utterance come from the other language, they are marked as @s. Thus, in an utterance marked as Cantonese using [- yue], there could be an English word marked as @s. However, in the other lines without the [- yue] marking, the @s indicates that the word is in Cantonese.

Once these markings are in place, MOR can be run twice on the corpus. In the first run, the English MOR is used and the –s"[- yue]" switch is used to exclude the Cantonese utterances. Then, English POST is run to disambiguate. After that, MOR must be switched to use the Cantonese MOR and the +s[- yue]" switch is used to analyze only the Cantonese utterances. Then, Cantonese POST is run to disambiguate the remaining undisambiguated forms. The result is a corpus tagged in both languages.

Processing of bilingual corpora can also be facilitated by special treatment of forms that play a role in both languages. For example, the English lexicon includes a set of Cantonese interjections and sentence final particles that are customarily used to embroider English sentences. Rather than treating these as code switching, we treat them as borrowings marked with the feature [lan yue]. Because TalkBank files use UTF-8 throughout, we

can enter these forms using Chinese characters in the middle of English sentences. Correspondingly, there are certain English words, such as "sorry" or "byebye" that are customarily used in the middle of Cantonese dialogs. These forms are entered into the Cantonese lexicon in a file called co-eng.cut. Yet another area of interlanguage usage involves proper nouns. In Cantonese utterances, we enter English proper nouns in roman characters beginning with capitals, because MOR treats words beginning with capitals as proper nouns by default. In the English files, we then use capitalized romanizations of Cantonese proper nouns. Thus, these cross-language proper nouns as not treated as borrowings. This is done, because we believe that proper nouns are language neutral, except for aspects of pronunciation, which would require further analysis.

11. Syntactic Dependency Analysis

Once tagging with MOR and POST is complete, we apply a deterministic grammatical relations tagger called GRASP (Sagae, Davis, Lavie, MacWhinney, & Wintner, 2010). GRASP constructs a %syn line with labeled GR arcs that describe the structure of sentences in terms of pairwise grammatical relations between words. These grammatical relations involve two dimensions: attachment and valency. In terms of attachment, each pair has a head and a dependent. Along the valency dimension, each pair has a predicate and an argument. Each dependency relation is labeled with an arc and the arc has an arrow which points from the predicate to argument. Valency relations open up slots for arguments. In English, modifiers (adjectives, determiners, quantifiers) are predicates whose arguments are the following nouns. In this type of dependency organization the argument becomes the head. However, in other grammatical relations, the predicate or governor is the head and the resultant phrase takes on its functions from the predicate. Examples of predicate-head GRs include the attachment of thematic roles to verbs and the attachment of adjuncts to their heads. Here is an example of the coding of the sentence *the big dog chased five cats* for dependencies:

```
*TXT:   the big dog chased five cats.
%mor:   det|the adj|big n|dog v|chase-PAST quant|five
        n|cat-PL.
%gra:   1|3|DET 2|3|MOD 3|4|SUBJ 4|0|ROOT
        5|6|QUANT 6|4|OBJ
```

This notation can be described in this way:

1. The determiner *the* is the first item and it attaches to the third item *dog*. Here the determiner is the predicate and the dependent. The GR here is DET or determination.
2. The adjective *big* is a predicate that attaches as a dependent of *dog*. The GR here is MOD or modification.
3. The noun *dog* is the head of the phrase *the big dog* and it attaches as a dependent subject or SUBJ of the predicate *chased*. Here we ignore the attachment of the suffix *–ed* to the verb.
4. The verb *chased* is the root of the clause. It attaches to the zero position of the "root" of the sentence.
5. The quantifier *five* attaches to the noun *cats* through the QUANT relation.
6. The noun *cats* attaches as a dependent to the verb *chased* through the OBJ or object relation.

GRASP for English uses 42 grammatical relations: 13 predicate-head relations, 16 argument-head relations, 4 links to the root node, 5 series relations, and 4 relations for punctuation. In addition, we use 13 GRs to mark types of ellipsis. For Chinese and Spanish only a few changes in these GRs are needed. However, Japanese uses eight additional relations.

12. Refinement

Testing for the accuracy of tagging by MOR, POST, and GRASP relies on the TRNFIX program for comparing newly created %mor and %syn lines with lines in the gold standard corpora. When mismatches are detected, the user can triple click the line and take a look at the original coding to understand the discrepancy. For POST, the developer of the grammar can list the internal contents of the statistical disambiguator with the POSTLIST program and can modify rules by hand using the POSTMODRULES program.

13. Analysis

Once high-quality, disambiguated %mor and %syn lines have been produced, the CLAN programs can be used to analyze morphosyntactic features in development. The possible methods here are quite extensive. Researchers usually need to spend 2-3 days learning the basic analysis methods provided in CLAN programs such as FREQ, MLU, VOCD, KWAL, COMBO, GEM, and so on.

In addition to these methods for custom analyses, we provide several forms of package analysis. For example, the MORTABLE program computes all frequencies of all grammatical categories on the %mor line across a collection of transcripts for direct opening as an Excel spreadsheet. The EVAL program computes a series of indices similar to those produced by the SALT program (Miller & Chapman, 1983) for clinical evaluations. It is also possible to conduct automatic computation of grammatical profile analyses in the DSS (Lee, 1974) or IPSyn (Sagae, Lavie, & MacWhinney, 2005) frameworks.

14. Conclusion

Recently, the tagged English corpora have been used in several tests of computational models of language acquisition (Borensztajn, Zuidema, & Bod, 2009; Freudenthal, Pine, & Gobet, 2010; Li, Zhao, & MacWhinney, 2007; Perfors, Tenenbaum, & Wonnacott, 2010; Waterfall, Sandbank, Onnis, & Edelman, 2010) and more such work is in progress. The emergence of this new line of work represents a major step forward for child language research. It will be interesting to see how this work develops as we manage to tag more corpora in more languages across the complete set of TalkBank data.

15. References

Beasley, K., & Karttunen, L. (2003). *Finite state morphology*. Stanford, CA: CSLI Publications.

Berlitz. (2005). *Berlitz Italian verbs handbook: 2nd Edition*. New York: Berlitz.

Borensztajn, G., Zuidema, W., & Bod, R. (2009). Children's grammars grow more abstract with age - Evidence from an automatic procedure for identifying the productive units of language. *Topics in Cognitive Science, 1*, 175-188.

Fredkin, E. (1960). Trie memory. *Communications of the ACM, 3*, 490-499.

Freudenthal, D., Pine, J., & Gobet, F. (2010). Explaining quantitative variation in the rate of Optional Infinitive errors across languages: A comparison of MOSAIC and the Variational Learning Model. *Journal of Child Language, 37*, 643-669.

Lee, L. (1974). *Developmental Sentence Analysis*. Evanston, IL: Northwestern University Press.

Li, P., Zhao, X., & MacWhinney, B. (2007). Dynamic self-organization and early lexical development in children. *Cognitive Science, 31*, 581-612.

MacWhinney, B. (2000). *The CHILDES Project: Tools for Analyzing Talk. 3rd Edition*. Mahwah, NJ: Lawrence Erlbaum Associates.

Miller, J., & Chapman, R. (1983). *SALT: Systematic Analysis of Language Transcripts, User's Manual*. Madison, WI: University of Wisconsin Press.

Perfors, A., Tenenbaum, J. B., & Wonnacott, E. (2010). Variability, negative evidence, and the acquisition of verb argument constructions. *Journal of Child Language, 37*, 607-642.

Sagae, K., Davis, E., Lavie, A., MacWhinney, B., & Wintner, S. (2010). Morphosyntactic annotation of CHILDES transcripts. *Journal of Child Language, 37*, 705-729.

Sagae, K., Lavie, A., & MacWhinney, B. (2005). Automatic measurement of syntactic development in child language *Proceedings of the 43rd Meeting of the Association for Computational Linguistics* (pp. 197-204). Ann Arbor: ACL.

Waterfall, H., Sandbank, B., Onnis, L., & Edelman, S. (2010). An empirical generative framework for computational modeling of language acquisition. *Journal of Child Language, 37*, 671-703.

Wintner, S. (2007). Finite-state technology as a programming environment. In A. Gelbukh (Ed.), *CICLing 2007, LNCS 4394* (pp. 97-106). Heidelberg: Springer.