Child Language Data Exchange System Tools for Clinical Analysis

Brian MacWhinney, Ph.D.[1] and Davida Fromm, Ph.D.[1]

ABSTRACT

The Child Language Data Exchange System Project has developed methods for analyzing many aspects of child language development, including grammar, lexicon, discourse, gesture, phonology, and fluency. This article will describe the methods available for each of these six fields, and how they can be used for assessment in the clinical setting.

KEYWORDS: Databases, production, gesture, lexicon, syntax

Learning Outcomes: As a result of this activity, the reader will be able to (1) summarize the ways in which language sample data can be recorded and transcribed in the clinical setting and (2) summarize the ways in which data transcribed in CHAT can be analyzed using CLAN.

Programs in speech-language pathology often recommend language sample analysis as a fundamental component for clinical training and practice. However, in real clinical situations, there is seldom enough time to collect, transcribe, and analyze speech samples in a meaningful way. As Heilmann observes, "Language sample analysis (LSA) is like flossing your teeth: it's something we all know we should do, but the majority of us neglect to do so on a regular basis."[1(p. 4)] However, Dunn and colleagues found that measures derived from language sample analysis do a better job at spotting language disorders than do standardized tests.[2]

Recently, the Child Language Data Exchange System (CHILDES) Project has begun to address this problem by creating new methods that can increase the speed and accuracy of transcription and provide fully automatic analysis, once an accurate transcript has been produced. The system for doing this is the KIDEVAL component of the CLAN programs. This article will begin with a description of recording methods, CLAN transcription, and the use of KIDEVAL. We will then review an additional series of areas for which automatic transcript analysis methods are not yet available, but which may still be relevant to clinical practice.

[1]Department of Psychology, Carnegie Mellon University, Pittsburgh, Pennsylvania.

Address for correspondence: Brian MacWhinney, Ph.D., Department of Psychology, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213 (e-mail: macw@cmu.edu).

Automating Child Speech, Language and Fluency Analysis; Guest Editor, Brian MacWhinney, Ph.D.

Semin Speech Lang 2016;37:63–73.
DOI: http://dx.doi.org/10.1055/s-0036-1580743.
ISSN 0734-0478.

RECORDING METHODS

Language sample analysis begins with the recording of a speech sample. Some clinicians may attempt to extract samples through direct transcription while listening to a child. However, this method is almost guaranteed to systematically miss important features of the child's productions. Moreover, a set of notes taken on the fly may not have sufficiently accurate data, if you decide to do some other type of analysis later. With modern digital recording technology, recording and playback of a 10-minute language sample is now quite easy. In the clinical setting, the simplest setup involves recording directly to a laptop computer through a high-quality USB microphone. Details regarding equipment for audio recording can be found at http://talkbank.org/info/da.html and for video recording at http://talkbank.org/info/dv.html.

TRANSCRIPTION METHODS

Once a 10-minute language sample has been extracted, a researcher or clinician can use the CLAN editor to create a transcript. Crucially, transcribing with the new CLAN programs that link audio or video to the transcript takes about half the time or less, compared with earlier methods. A short version of the CLAN manual—*A Clinician's Complete Guide to CLAN and Praat* by Nan Bernstein Ratner and Shelley Brundage—can be downloaded from http://childes.talkbank.org/manuals/clin-CLAN.pdf.[3] This GUIDE can be used in combination with a self-guiding tutorial system downloadable from http://childes.talkbank.org/clan/tutorial.zip.

New users should familiarize themselves right away with the use of the CLAN text editor. This editor operates much like Notepad on Windows (Microsoft, Redmond, Washington) or TextEdit on MacOSX (Apple Computer, Cupertino, CA). In other words, it is a barebones version of MS Word (Microsoft). When you start it up, CLAN opens a Commands Window. To open an editor window, you can type control-O, as in MS Word. The CLAN editor has all the same basic commands as MS Word, without of the many additional bells and whistles. However, it differs from MS Word in the fact that it creates a text-only file that can also be opened by a wide variety of other editors without conversion. In addition, the CLAN editor provides methods for checking the accuracy of transcription and for linking transcripts to audio that are not available in MS Word. For these reasons, transcribers are advised to rely on the CLAN editor rather than MS Word when transcribing a language sample.

TRANSCRIPTION CODES

To derive the maximum value from CLAN's facilities for automatic transcript analysis and to facilitate rapid analysis, the transcription must provide the adult target form for each word that the child produces. Because the transcript is linked to the audio, it is always possible to conduct detailed phonological analysis later on through use of the PHON program (https://www.phon.ca/phontrac), which is compatible with CLAN. There are several methods available to guarantee that transcripts provide standard forms. For example, nonwords should be preceded with an ampersand, as in *&gaga*. Interjections and communicators are given standard forms such as *oughtoh* and *okay*. Omitted segments are marked with parentheses, as in *(be)cause*. Repetition and retraces are also marked with special characters. The full set of these conventions can be found in the *Clinician's Guide to CLAN*. It helps to keep in mind the fact that the goal of these various transcription conventions is to provide the automatic analyzers with a series of standard words in English.

LINKAGE

Linkage of a transcript to the media is a helpful and interesting facility provided by CLAN. Although linkage is not required for the automatic analyses produced by the KIDEVAL program described below, there are five reasons for linking transcripts to media. First, transcription that occurs along with linkage is actually faster than transcription without linkage. Second, you may want to refine the accuracy of a transcript and this is much easier to do when segments can be replayed through a mouse click. Third, you may want to use a transcript as a teaching or learning device. Fourth, you

may want to use a transcript with parents to explain to them the status of a child's language in clearer terms. Fifth, you may wish to contribute the transcripts you create to the CHILDES database. In that case, we would want to have the transcripts linked to media.

The CLAN Editor provides five methods for linking transcripts to audio or video. These links are contained in little bullet symbols that are placed at the end of each line of the transcript. Inside the links are the time values in milliseconds for the beginning and end of the related utterance in the media file. The various methods for creating links are described in fullest detail in the CLAN manual. For people who are just beginning with learning how to link in CLAN, perhaps the easiest method involves first using the Walker Controller system to create a transcript without media links, and then adding the links using the F5 Transcriber method. These approaches are explained in the *Clinician's Guide*, the tutorial, and the main CLAN manual.

PREPARATION FOR AUTOMATIC ANALYSIS

Once a transcript has been created in CLAN, it can be analyzed automatically using the KIDEVAL program. This program joins together a series of more basic CLAN analyses into a single, coherent package for smoother clinical analysis. Some of the analyses produced in KIDEVAL are based on patterns that can be derived from the main orthographic line. Others depend on analyses that require prior automatic computation of the %mor tier for morphosyntactic analysis and the %gra tier for syntactic analysis. Before describing the functioning of KIDEVAL, we need to consider how to add %mor and %gra tiers to a transcript to allow for automatic analysis using KIDEVAL. To do this, we need to review the functioning of these three commands: MOR, POST, and MEGRASP.

PREPARATION FOR RUNNING MOR

Before running MOR, you should first verify that all the words in your transcript will be recognized by MOR. To do this, you need to run this command:

```
mor +xb *.cha
```

The asterisk in this command is a wild card representing any name. If you have a folder full of files such as 01.cha, 02.cha, 03.cha, this command will run on all of them to locate any unrecognized words. If you have only one file to analyze, such as oscar.cha, then the command would be:

```
mor +xb oscar.cha
```

The output from this analysis will go to a file called oscar.ulx.cex. If that file is empty, it means that all words are recognized and you can proceed. If some words are not recognized, you can double-click on the lines indicating the missing items. CLAN will then open up the position in the original file where the unrecognized word appears. You will need to either correct the spelling of that word or else add that word to the ENG grammar as a missing word. For example, if your transcript has the word *unctuous,* ENG will not recognize it and you will need to create a file with a name like additions. cut into which you put this line

```
unctuous {[scat adj}]
```

and then you can put that file into the /lex folder of the ENG grammar. In this way, you can add missing words to the ENG lexicon.

HOW MOR WORKS

The MOR program uses a set of morphological rules and lexical forms called a MOR grammar to create a new line called the %mor tier. Here is an example of a main line in CHAT from the eve15.cha file in the Brown corpus, along with a %mor line that has been computed through the operation of the MOR program.

```
*CHI: see if I can blow it up.
%mor: v|see^co|see conj|if pro:sub|I
mod|can^n|can v|blow^n|blow pro|it
prep|up^adv|up
```

In this example, several of the words are ambiguous in terms of their parts of speech. Specifically, the word *see* could be either a verb

or a communicator, *can* could be either a modal auxiliary or a noun, *blow* could be a noun or a verb, and *up* could be a preposition or an adverb. Fortunately, these ambiguities can all be resolved by the POST program, which runs automatically after MOR by default. This means that, if all the words in your transcript are recognized, MOR (followed automatically by POST) can run over dozens of files automatically and produce accurate output in seconds. Here is the result for this sentence after POST has run on the output from MOR:

*CHI: see if I can blow it up.
%mor: v|see conj|if pro:sub|I mod|
can v|blow pro|it adv|up.

This result is correct. In general, the accuracy rate for disambiguation by POST is ~97%.

RUNNING MOR, POST, AND MEGRASP

Once you have run MOR with the +xb and you know that all words will be recognized, you should run it across your files, using this command:

mor *.cha +1

This will run MOR and POST and the results should be disambiguated. After the %mor line has been computed, the next step is to run MEGRASP. This step is not necessary for all analyses, but the running of MEGRASP is so automatic, that it makes sense to simply run it in all cases, once MOR and POST are finished. Running MEGRASP produces this additional output:

*CHI: see if I can blow it up.
%mor: v|see conj|if pro:sub|I mod|
can v|blow pro|it adv|up.
%gra: 1|0|ROOT 2|5|LINK 3|5|
SUBJ 4|5|AUX 5|1|CJCT 6|5|OBJ 7|5|
JCT 8|1|PUNCT

To see the structure coded by the %gra line, you can double click on that line and (if you are connected to the Internet) CLAN will run a web service program from the servers at Carnegie Mellon University (CMU) that will create this dependency graph for the sentence in Fig. 1.

Fig. 1 displays the syntactic structure of the child's utterance in terms of a set of grammatical relations holding between words. The nature of these grammatical relations is explained in the CLAN manual. For a fuller understanding of the principles of Dependency Grammar analysis, please consult Kübler et al.[4]

The complete chain of commands needed to produce the %mor and %gra line is this:

mor *.cha +1

check *.cha

megrasp *.cha +1

check *.cha

You type these commands into CLAN's Commands Window one by one. When running these commands, you need to set your MOR LIB to the folder that includes the complete English MOR grammar, called ENG. You can download ENG from childes.talkbank.org/morgrams. There you will also find MOR grammars for ~10 other languages, including Cantonese, Chinese, Danish, Dutch, French,

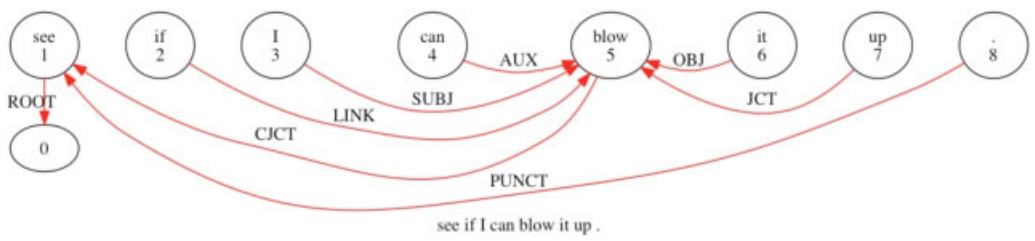

Figure 1 Dependency graph created by triple-clicking.

German, Hebrew, Japanese, Italian, and Spanish. The availability of MOR for all these languages means that these tools can be used with well over half of the children in the world, as well as several bilingual configurations. However, if you are only working with English data, you only need to download the ENG grammar.

When running the four commands above, make sure that you first make a copy of the folder containing your original files, because the addition of the +1 switch to the commands overwrites the originals. Also, note that the second and fourth commands run the CHECK program on your outputs to make sure that no words are either missing or still ambiguous.

RUNNING KIDEVAL

KIDEVAL uses the information on the %mor tier and the main line to compute a variety of clinically-relevant indices, which we will review below. However, at this point, you should realize that the actual running of KIDEVAL is an easy next step. You simply type:

kideval +t*CHI +leng *.cha

The +t*CHI switch is needed to tell KIDEVAL to only focus on the child. The +leng switch is needed to tell KIDEVAL that the language is English. This is because KIDEVAL can also work for other languages that have a MOR grammar. The result of this command is a tab-delimited text file that can be opened and saved as a Microsoft Excel spreadsheet. Excel may complain about opening the file not being trusted, but you should just tell Excel to open it anyway. In the spreadsheet, the rows represent each of the transcript files being analyzed and the columns indicate the variables or measures being calculated. Currently, KIDEVAL computes these 26 measures:

1. Total Utts: This is the total number of utterances can be computed from the main line.
2. MLU Utts: This is the number of utterances used for computing mean length of utterance (MLU), which Brown defines as excluding utterances with unrecognized words.[5]
3. MLU Words: This is the mean length in words of the utterances included in the MLU count.
4. MLU Morphemes: This is the mean length in morphemes, which must be computed from the %mor line.
5. MLU 100 Utts: This is a count of the first 100 utterances. It will be 100, unless there are not enough utterances.
6. MLU 100 Words: This is the MLU of the first 100 child utterances in words.
7. MLU 100 Morphemes: This is the MLU of the first 100 child utterances in morphemes, based on the %mor line.
8. FREQ types: This is the total word types in all the child's utterances.
9. FREQ tokens: This is the total word tokens.
10. FREQ TTR: This is the type/token ratio (TTR).
11. Verbs/Utt: This is the verbs per utterance. This can be less than 1.0 for young children.
12. TD Words: This is the total number of words for each speaker, as used for computing TIMEDUR. Note that may be different from the total number of word tokens used for computing FREQ.
13. TD Utts: This is the total number of utterances for each speaker, as used for computing TIMEDUR.
14. TD Time: This is the total duration in seconds of utterances for each speaker. Computing this depends on the presence of bullets with time values.
15. TD Words/Time: This is the words per second, if time values have been created.
16. TD Utts/Time: This is the utterances per second, if time values have been created.
17. Word Errors: This is the number of words involved in errors, as marked by [*] indicators.
18. Utt Errors: This is the number of utterances involved in errors/.
19. Retracing [//]: This is the number of retracings, as marked by [//].
20. Repetition [/]:This is the number of repetitions, as marked by [/].

21. DSS Utterances: This is the number of utterances available for computation of the Developmental Sentence Score (DSS).
22. DSS Score: This is the score derived from the complete DSS analysis, except for the sentence point.
23. vocD (vocabulary diversity) score: This is the score derived from the vocD vocabulary density analysis.
24. IPSyn Utterances: This is the utterances used in computing Index of Productive Syntax (IPSyn).
25. IPSyn Score: This is the IPSyn score.
26. This is the frequencies of each of Brown's 14 grammatical morphemes.

Most of these measures are self-explanatory, and most can be derived from either the main line or the %mor line. However, the five measures that measure time values (12 to 16) require that the transcript include bullets marking the linkage of utterances to begin and end times in the media. Computation of vocD is based on stems in the %mor line by default, and it conducts multiple automatic sampling passes through the data, reporting the average across these passes. Computation of DSS and IPSyn is automatic, but it depends on some additional files, which we will describe below.

It is possible to control the operation of KIDEVAL by editing the script files for individual languages that are stored in the /lib/kideval folder. To use one of these files, just add its name to the +l switch, as in +lfra to use the fra.cut file for French. If you use the switch +leng or +lfra, then the program relies on a built-in script file, instead of your user-defined file.

Within the script file, each line defines a type of search string. For example, this line searches for all instances of masculine singular marking in French:

+&m,&sg +&m,-sg "m-sg"

Items separated by a comma are treated as AND; items separated by a space are treated as OR. To include combinations of morphemes in a KIDEVAL spreadsheet, you must run a separate FREQ program, such as this one that looks for adjective + noun or noun + adjective combinations in French:

freq +s"@|adj @|n" +s"@|n @|adj" +d2 *.cha

This command will create an Excel output structured like that for KIDEVAL and you may wish to cut and paste the relevant columns from that output into your overall KIDEVAL spreadsheet.

After the first 25 measures, KIDEVAL then includes 14 more columns that report the frequencies of usage for the 14 morphemes studied by Brown.[5] Usage of these morphemes in obligatory contexts is often taken as an indicator of the child's level of linguistic development. For other languages, a very different set of morphemes will be relevant, as specified in the script file for each language.

KIDEVAL AND EVAL

CLAN also includes a program called EVAL that serves as the equivalent of KIDEVAL for the study of adult aphasia.[6] However, unlike KIDEVAL, the EVAL program allows the clinician or researcher to compare a transcript from a given participant with a larger group of clinically similar transcripts, all collected using a common protocol. For example, one can compare a person who appears to have Broca's aphasia with the transcripts from 94 other people with Broca's aphasia in the AphasiaBank database. This comparison yields raw scores and standard deviations across each of the 25 measures in KIDEVAL, along with a few additional measures. It is our intention to configure KIDEVAL to work in this way too. We will do this by analyzing groups of transcripts in the current CHILDES database from normally developing children in 6-month segments. Although CHILDES data were not collected using a standard protocol, we will group together similar data collection situations, such as toy play or interview questions.

BEYOND KIDEVAL

In the clinical setting, it is important to achieve as much automation of transcription and transcript analysis as possible. Currently, the combined use of linked transcription, automatic MOR/POST/MEGRASP, and KIDEVAL

can shave hours off the work of clinical transcript analysis. Moreover, once KIDEVAL is configured to include comparison groups to the larger CHILDES database, it will be easier to judge how a child stands in relation to others of their age and social situation.

Although we are interested in maximizing the power of automatic analysis, there are still many aspects of language development that could be important for clinical analysis that are not yet automated, but which can be computed through CLAN. In the next sections, we will review these additional analysis types for the areas of grammar, lexicon, discourse, gesture, phonology, and fluency.

GRAMMATICAL DEVELOPMENT—DSS AND IPSYN

The study of grammatical development is significant not just for linguistic and psycholinguistic theory, but also for clinical practice. The diagnosis and treatment of specific language impairment hinges closely on the ability to compare children's learning of grammatical markings[7,8] with a normal standard. Evaluation of progress in grammar also plays a major role in understanding language in autism,[9,10] stuttering,[11] early focal lesions,[12] Williams syndrome,[13] and other developmental disabilities.

One of the major goals of language sample analysis has been the characterization of a child's developmental level through a measure that provides an overall linguistic profile. The three most often used measures of this type are the DSS,[14] the Index of Productive Syntax,[15] and the Language Assessment, Remediation, and Screening Profile.[16] KIDEVAL currently includes methods for computing DSS and IPSyn automatically, but not Language Assessment, Remediation, and Screening Profile.

If the clinician is interested in seeing not just a summary score for DSS and IPSyn, then they can run each of these analyses independently outside of KIDEVAL. The program will still run automatically, but the output will be much more complete. For DSS, the output will take each of the 50 sentences used in the analysis and provide a summary of the points assigned across each of the eight grammatical areas (indefinite pronouns, personal pronouns, main verbs, secondary verbs, negatives, conjunction, interrogative reversal, and wh-question). Here is example output for the first five sentences in a transcript:

Sentence	IP	PP	MV	SV	NG	CNJ	IR	WHQ	S	TOT
I like this.	1	1	1						1	4
I like that.	1	1	1						1	4
I want hot dog.		1	1						0	2
I like it .	1	1	1						1	4
what this say.	1	-			-	2		0		3

Developmental Sentence Score: 4.2

The final DSS score in this example (4.2) is simply the average score for each of the sentences being coded. The rules used by CLAN's DSS program can be viewed in the file called eng.cut in the lib/dss folder in the CLAN package. There is also a version of DSS for Japanese, but none yet for other languages. Full computation of DSS requires hand entry of a sentence point (the ninth column in the example). When DSS is run fully automatically, this point is computed by excluding sentences with error markings and missing words. As a result, it is only partially accurate in the fully automatic version.

IPSyn relies on a more complex set of syntactic patterns. However, as with DSS, these can be computed from the %mor line without relying on patterns on the %gra syntactic line. The rules for IPSyn can be found in /ipsyn/eng.cut in the CLAN package. Recent work by Lubetich and Sagae[17] and Sahakian and Snyder[18] shows that it may be possible to exceed the diagnostic accuracy of measures such as DSS and IPSyn by using data-driven methods for combining features on both the %mor and %syn lines in CHAT files. Once we have tested these more powerful methods, we will include them in KIDEVAL, along with DSS and IPSyn.

LEXICAL DEVELOPMENT

MLU, DSS, and IPSyn and the various morpheme counts provided by KIDEVAL provide

the clinician with measures of the growth of the child's morphosyntactic abilities. However, much of language development involves the acquisition of the lexicon, including words, meanings, collocations, and idioms. In fact, by age 4, the core syntactic and morphological structures are well in place and further language development often focuses on enhancements to lexical and discourse structures.

CLAN provides several methods for studying and assessing lexical development. These include:

1. FREQ. This program provides a wide range of methods for tracking the development of lexical frequency for the whole vocabulary, individual words, semantic groups, or parts of speech either across children or across samples from the same child.
2. TTR. The FREQ program also computes the TTR, which is simply the ratio of the number of different word types in a transcript over the total number of words (tokens) being used. TTR has often been used as a measure of lexical diversity, despite the fact that it is only reliable for sample sizes larger than those collected in clinical practice.
3. vocD. To correct for the tendency of TTR to overestimate lexical diversity in small samples, Malvern et al created the vocD or vocabulary diversity measure.[19] This measure uses repeated Monte Carlo sampling to estimate the stability of the D statistic. CLAN includes the original code used by Malvern et al for computing vocD but adds the capacity to compute vocD based on either the full word forms found in the main line or the word stems found in the %mor line. Although this measure avoids some of the problems facing TTR, it requires at least 100 utterances for reliable estimation.[20]
4. Moving Average Type-Token Ratio (MATTR). Two other more recent approaches to measuring vocabulary diversity are the MATTR[21] and the Measure of Textual Lexical Diversity (MTLD).[22] Comparing MATTR, MTLD, TTR, vocD, and HD-D,[23] Fergadiotis et al concluded that MATTR and MTLD computed the most reliable estimates of vocabulary diversity without influence from sample size or text genre.[20] Of these two measures, MATTR is the one that is most intuitively interpretable, because it relates directly to the traditional TTR but differs from that ratio by computing across a moving average window that allows for relative independence from sample size. As a result, this is the measure that we now prefer within EVAL and KIDEVAL.
5. COOCCUR. To acquire full control of the adult lexicon, children need to learn not only tens of thousands of words, but also an equal number of word combinations in terms of collocations and idioms. To trace children's ongoing learning of word combinations, CLAN provides the COOCCUR program, which outputs the frequencies of all combinations involving N words, where N can be any number set by the user, such as 2, 3, 4 or more. In addition, the KWAL (Key Word and Line) program can be used to create a corpus that shows a key word in all of its relevant contexts.

DISCOURSE DEVELOPMENT

Clinical language assessment for children over age 5 should pay particular attention to the development of narrative and discourse skills. However, methods for achieving this assessment remain largely underdeveloped. One method recently introduced into CLAN for this computation for the computation of propositional density. Propositional density analyses are based on a conceptual framework provided by Kintsch and Van Dijk,[24] as implemented computationally by Covington and colleagues in the CPDIR program.[25] We have reimplemented that analysis inside CLAN, and this reimplementation matches up nearly perfectly with Covington's CPDIR. CPDIR has been used to document the extent to which nuns who used a high propositional density in their early diary writings were less likely to develop dementia in their later years.[26]

In addition to CPDIR, CLAN can assess discourse structure through automatic computation of MLU, mean length of turn, and a

variety of measures using the KEYMAP, CHAINS, and CHIP programs.

GESTURE DEVELOPMENT

Researchers studying gesture have typically processed AphasiaBank files using the ELAN program developed in 2002 by the Max-Planck Institute for Psycholinguistics in Nijmegen. Analysis through ELAN can be facilitated by the use of the CHAT2ELAN and ELAN2-CHAT commands in CLAN that work to convert to and from ELAN. ELAN is an excellent program for gesture analysis. However, gesture analysis with ELAN is extremely time-consuming and not well adapted to the clinical context. CLAN provides simpler, alternative ways of notating gesture directly in a transcript. One method relies on codes such as &=points:phone for the notation of a gesture pointing to the stove. This type of notation indicates that a gesture occurs at a particular place in the middle of a verbal utterance. This type of gesture is very common for young children. Older children will often construct longer gesture chains. In such cases, CLAN provides methods for linking detailed structural analyses of gesture chains back to the basic transcript. We hope that future work with gesture makes wider use of some of these facilities.

PHONOLOGICAL DEVELOPMENT

The clinical analysis of phonological development typically relies on administration of articulation tests such as the Goldman-Fristoe or the Arizona Articulation Proficiency Scale.[27,28] Although these tests are quick to administer, they fail to cover many aspects of phonological development. Within the context of the PhonBank Project,[29] which is one of the TalkBank databases, we have developed the PHON program for computerized analysis of phonological patterns in language samples. Although many aspects of PHON have been automated, such as the insertion of the target phonology and syllabic segmentation, the basic work of transcription into IPA has not been automated. However, future versions of PHON will include automatic computation of a wide range of variables of importance to clinicians, including PROPH^{+},[30,31] pMLU,[32] and index of phonetic complexity (IPC).[33]

FLUENCY DEVELOPMENT

CHAT transcription provides a variety of special markers designed to encode and analyze disfluencies,[34] including fillers, initial repetition, blocking, internal pausing, drawling, word repetition, and retracing. Utterance internal and external pause time can be computed using the TIMEDUR program. Currently, pause length must be marked manually. However, we are developing word-level alignment methods that will facilitate automatic computation of pause duration. For acoustic analysis of pauses and dysfluencies, transcripts in CHAT can be automatically converted to the format needed for the Phon program,[29] which links tightly with acoustic analysis in Praat.[35]

CONCLUSION

The CLAN programs provide a wide range of powerful tools for language sample analysis. Clinicians will want to focus on the use of the automatic assessments provided by coding a transcription with MOR/POST/MEGRASP and then running KIDEVAL. This package provides a wide range of basic assessments of grammar and lexicon. However, clinicians and researchers interested in further details analysis can also use separate CLAN programs to analyze further aspects of grammar and lexicon, along with features of discourse, gesture, phonology, and fluency. Together these new resources for clinical analysis suggest that language sample analysis should become an increasingly important component of clinical assessment.

DISCLOSURES

Brian MacWhinney: no conflicts.
Davida Fromm: no conflicts.

ACKNOWLEDGMENTS

This work was supported by grant HD082736 from the National Institutes of Health (NICHD).

REFERENCES

1. Heilmann J. Myths and realities of language sample analysis. Perspect Lang Learn Educ 2010;17:4–8
2. Dunn M, Flax J, Sliwinski M, Aram D. The use of spontaneous language measures as criteria for identifying children with specific language impairment: an attempt to reconcile clinical and research incongruence. J Speech Hear Res 1996;39(3):643–654
3. Bernstein Ratner N, Brundage SA. Clinician's Complete Guide to CLAN and Praat. Available at: http://childes.talkbank.org/manuals/clin-CLAN.pdf. 2015. Accessed March 24, 2016
4. Kübler S, McDonald R, Nivre J. Dependency Parsing. San Rafael, CA: Morgan and Claypool; 2009
5. Brown R. A First Language: The Early Stages. Cambridge, MA: Harvard; 1973
6. MacWhinney B, Fromm D. AphasiaBank as big data. Semin Speech Lang 2016;37(1):10–22
7. Rice ML, Wexler K. Toward tense as a clinical marker of specific language impairment in English-speaking children. J Speech Hear Res 1996;39(6):1239–1257
8. van der Lely HK. Domain-specific cognitive systems: insight from Grammatical-SLI. Trends Cogn Sci 2005;9(2):53–59
9. Kjelgaard MM, Tager-Flusberg H. An investigation of language impairment in autism: implications for genetic subgroups. Lang Cogn Process 2001;16(2–3):287–308
10. Tager-Flusberg H, Paul R, Lord C. Language and communication in autism. In: Volkmar F, Paul R, Klin A, Cohen D eds. Handbook of Autism and Pervasive Developmental Disorders. Hoboken, NJ: Wiley; 2005:335–364
11. Howell P. Recovering from Stuttering. New York: Psychology Press; 2011
12. Booth JR, MacWhinney B, Thulborn KR, Sacco K, Voyvodic JT, Feldman HM. Developmental and lesion effects in brain activation during sentence comprehension and mental rotation. Dev Neuropsychol 2000;18(2):139–169
13. Karmiloff-Smith A, Brown JH, Grice S, Paterson S. Dethroning the myth: cognitive dissociations and innate modularity in Williams syndrome. Dev Neuropsychol 2003;23(1–2):227–242
14. Lee L. Developmental Sentence Analysis. Evanston, IL: Northwestern University Press; 1974
15. Scarborough HS. Index of productive syntax. Appl Psycholinguist 1990;11:1–22
16. Crystal D, Fletcher P, Garman M. The grammatical analysis of language disability. 2nd ed. London, UK: Cole and Whurr; 1989
17. Lubetich S, Sagae K. Data-driven measurement of child language development with simple syntactic templates. Paper presented at: COLING2014, 2014; Dublin, Ireland
18. Sahakian S, Snyder B. Automatically learning measures of child language development. Paper presented at: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, 2012
19. Malvern D, Richards B, Chipere N, Purán P. Lexical Diversity and Language Development. New York, NY: Palgrave Macmillan; 2004
20. Fergadiotis G, Wright HH, Green SB. Psychometric evaluation of lexical diversity indices: assessing length effects. J Speech Lang Hear Res 2015;58(3):840–852
21. Covington MA, McFall JD. Cutting the Gordian knot: the moving-average type–token ratio (MATTR). J Quant Linguist 2010;17(2):94–100
22. McCarthy PM, Jarvis S. MTLD, vocd-D, and HD-D: a validation study of sophisticated approaches to lexical diversity assessment. Behav Res Methods 2010;42(2):381–392
23. McCarthy PM, Jarvis S. Voc-D: a theoretical and empirical evaluation. Lang Test 2007;24:459–488
24. Kintsch W, Van Dijk T. Toward a model of text comprehension and production. Psychol Rev 1978;85:363–394
25. Brown C, Snodgrass T, Kemper SJ, Herman R, Covington MA. Automatic measurement of propositional idea density from part-of-speech tagging. Behav Res Methods 2008;40(2):540–545
26. Kemper S, LaBarge E, Ferraro FR, Cheung H, Cheung H, Storandt M. On the preservation of syntax in Alzheimer's disease. Evidence from written sentences. Arch Neurol 1993;50(1):81–86
27. Goldman R, Fristoe M. The Goldman-Fristoe Test of Articulation–2. San Antonio, TX: Pearson Assessments; 2000
28. Fudala JB, Reynolds WM. Arizona Articulation Proficiency Scale: Manual. Torrance, CA: Western Psychological Services; 1986
29. Rose Y, MacWhinney B. The PhonBank Project: data and software-assisted methods for the study of phonology and phonological development. In: Durand J, Gut U, Kristoffersen G eds. The Oxford Handbook of Corpus Phonology. Oxford, UK: Oxford University Press; 2014:380–401
30. Computerized Profiling [computer program] . Version 9.7.0. Cleveland, OH: Case Western Reserve University; 2006
31. Computerized profiling (PROPH+) [computer program]. Version 9.7.0. San Antonio, TX: The Psychological Corporation; 1993
32. Ingram D. The measurement of whole-word productions. J Child Lang 2002;29(4):713–733
33. Nelson LK, Bauer HR. Speech and language production at age 2: evidence for tradeoffs between linguistic and phonetic processing. J Speech Hear Res 1991;34(4):879–892

34. Bernstein Ratner N, Rooney B, MacWhinney B. Analysis of stuttering using CHILDES and CLAN. Clin Linguist Phon 1996;10(3): 169–188

35. Boersma P, Weenink D. Praat, a System for Doing Phonetics by Computer. Amsterdam, the Netherlands: Institute of Phonetic Sciences of the University of Amsterdam; 1996