Challenges in Automating Maze Detection

Eric Morley
CSLU
OHSU
Portland, OR 97239
morleye@gmail.com

Anna Eva Hallin
Department of Communicative
Sciences and Disorders
New York University
New York, NY
ae.hallin@nyu.edu

Brian Roark
Google Research
New York, NY 10011
roarkbr@gmail.com

Abstract

SALT is a widely used annotation approach for analyzing natural language transcripts of children. Nine annotated corpora are distributed along with scoring software to provide norming data. We explore automatic identification of *mazes* – SALT's version of disfluency annotations – and find that cross-corpus generalization is very poor. This surprising lack of cross-corpus generalization suggests substantial differences between the corpora. This is the first paper to investigate the SALT corpora from the lens of natural language processing, and to compare the utility of different corpora collected in a clinical setting to train an automatic annotation system.

1 Introduction

Assessing a child's linguistic abilities is a critical component of diagnosing developmental disorders such as Specific Language Impairment or Autism Spectrum Disorder, and for evaluating progress made with remediation. Structured instruments ("tests") that elicit brief, easy to score, responses to a sequence of items are a popular way of performing such assessment. An example of a structured instrument is the CELF-4, which includes nineteen multi-item subtests with tasks such as object naming, word definition, reciting the days of the week, or repeating sentences (Semel et al., 2003). Over the past two decades, researchers have discussed the limitations of standardized tests and how well they tap into different language impairments. Many have advocated the potential benefits of language sample analysis (LSA) (Johnston, 2006; Dunn et al., 1996). The analysis of natural language samples may be particularly beneficial for language assessment in ASD, where pragmatic and social communication issues are paramount yet may be hard to assess in a conventional test format (Tager-Flusberg et al., 2009).

At present, the expense of LSA prevents it from being more widely used. Heilmann (2010), while arguing that LSA is not too time-consuming, estimates that each minute of spoken language takes five to manually transcribe and annotate. At this rate, it is clearly impractical for clinicians to perform LSA on hours of speech. Techniques from natural language processing could be used to build tools to automatically annotate transcripts, thus facilitating LSA.

Here, we evaluate the utility of a set of annotated corpora for automating a key annotation in the de facto standard annotation schema for LSA: the Systematic Analysis of Language Transcripts (SALT) (Miller et al., 2011). SALT comprises a scheme for coding transcripts of recorded speech, together with software that tallies these codes, computes scores describing utterance length and error counts, among a range of other standard measures, and compares these scores with normative samples. SALT codes indicate bound morphemes, several types of grammatical errors (for example using a pronoun of the wrong gender or case), and *mazes*, which are defined as "filled pauses, false starts, and repetitions and revisions of words, morphemes and phrases" (Miller et al., 2011, p. 48).

Mazes have sparked interest in the child language disorders literature for several reasons. They are most often analyzed from a language processing perspective where the disruptions are viewed as a consequence of monitoring, detecting and repairing language, potentially including speech errors (Levelt, 1993; Postma and Kolk, 1993; Rispoli et al., 2008). Several studies have found that as grammatical complexity and utterance length increase, the number of mazes increases in typically developing children and children with language impairments (MacLachlan and

Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality, pages 69–77,
Baltimore, Maryland USA, June 27, 2014.

Chapman, 1988; Nippold et al., 2008; Reuterskiöld Wagner et al., 2000; Wetherell et al., 2007). Mazes in narrative contexts have been shown to differ between typical children and children with specific language impairment (MacLachlan and Chapman, 1988; Thordardottir and Weismer, 2001), though others have not found reliable group differences (Guo et al., 2008; Scott and Windsor, 2000). Furthermore, outside the potential usefulness of looking at mazes in themselves, mazes always have to be detected and excluded in order to calculate other standard LSA measures such as mean length of utterance and type or token counts. Mazes also must be excluded when analyzing speech errors, since some mazes are in fact self-corrections of language or speech errors.

Thus, automatically delimiting mazes could be clinically useful in several ways. First, if mazes can be automatically detected, standard measures such as token and type counts can be calculated with ease, as noted above. Automatic maze detection could also be a first processing step for automatically identifying errors: error codes cannot appear in mazes, and certain grammatical errors may be easier to identify once mazes have been excised. Finally, after mazes have been identified, further analysis of the mazes themselves (e.g. the number of word in mazes, and the placement of mazes in the sentence) can provide supplementary information about language formulation abilities and word retrieval abilities (Miller et al., 2011, p. 87-89).

We use the corpora included with the SALT software to train maze detectors. These are the corpora that the software uses to compute reference counts. These corpora share several characteristics we expect to be typical of clinical data: they were collected under a diverse set of circumstances; they were annotated by different groups; the annotations ostensibly follow the same guidelines; and the annotations were not designed with automation in mind. We will investigate whether we can extract usable generalizations from the available data, and explore how well the automated system performs, which will be of interest to clinicians looking to expedite LSA.

2 Background

Here we provide an overview of SALT and maze annotations. We are not aware of any attempts to automate maze detection, although maze detection closely resembles the well-established task of *edited word detection*. We also provide an overview of the corpora included with the SALT software, which are the ones we will use to train maze detectors.

2.1 SALT and Maze Annotations

The approach used in SALT has been in wide use for nearly 30 years (Miller and Chapman, 1985), and now also exists as a software package[1] providing transcription and coding support along with tools for aggregating statistics for manual codes over the annotated corpora and comparing with age norms. The SALT software is not the focus of this investigation, so we do not discuss it further.

Following the SALT guidelines, speech should be transcribed orthographically and verbatim. The transcript must include and indicate: the speaker of each utterance, partial words or stuttering, overlapping speech, unintelligible words, and any non-speech sounds from the speaker. Even atypical language, for example neologisms (novel words) or grammatical errors (for example ‘her went’) should be written as such.

There are three broad categories of SALT annotations: indicators of 1) certain bound morphemes, 2) errors, and 3) *mazes*. In general, verbal suffixes that are visible in the surface form (for example -ing in “going”) and clitics that appear with an unmodified root (so for example -n’t in “don’t”, but not the -n’t in “won’t”) must be indicated. SALT includes various codes to indicate grammatical errors including, but not limited to: overgeneralization errors (“goed”), extraneous words, omitted words or morphemes, and inappropriate utterances (e.g. answering a yes/no question with “fight”). For more information on these standard annotations, we refer the reader to the SALT manual (Miller et al., 2011).

Here, we are interested in automatically delimiting mazes. In SALT, filled pauses, repetitions and revisions are included in the umberella term “mazes” but the manual does not include definitions for any of these categories. In SALT, mazes are simply delimited by parentheses; they have no internal structure, and cannot be nested. Contiguous spans of maze words are delimited by a single set of parentheses, as in the following utterance:

(1) (You have you have um there/’s only)
there/’s ten people

[1] http://www.saltsoftware.com/

To be clear, we define the task of automatically applying maze detections as taking unannotated transcripts of speech as input, and then outputting a binary tag for each word that indicates whether or not it is in a maze.

2.2 Edited Word Detection

Although we are not aware of any previous work on automating maze detection, there is a well-established task in natural language processing that is quite similar: edited word detection. The goal of edited word detection is to identify words that have been revised or deleted by the speaker, for example ‘to Dallas’ in the utterance ‘I want to go to Dallas, um I mean to Denver.’. Many investigations have approached edited word detection from what Nakatani et al. (1993) have termed ‘speech-first’ perspective, meaning that edited detection is performed with features from the speech signal in addition to a transcript. These approaches, however, are not applicable to the SALT corpora, because they only contain transcripts. As a result, we must adopt a *text-first* approach to maze detection, using only features extracted from a transcript.

The text-first approach to edited word detection is well established. One of the first investigations taking a text-first approach was conducted by Charniak and Johnson (2001). There, they used boosted linear classifiers to identify edited words. Later, Johnson and Charniak (2004) improved upon the linear classifiers’ performance with a tree adjoining grammar based noisy channel model. Zwarts and Johnson (2011) improve the noisy channel model by adding in a reranker that leverages features extracted with the help of a large language model.

Qian and Liu (2013) have developed what is currently the best-performing edited word detector, and it takes a text-first approach. Unlike the detector proposed by Zwarts and Johnson, Qian and Liu’s does not rely on any external data. Their detector operates in three passes. In the first pass, filler words (‘um’, ’uh’, ‘I mean’, ’well’, etc.) are detected. In the second and third passes, edited words are detected. The reason for the three passes is that in addition to extracting features (mostly words and part of speech tags) from the raw transcript, the second and third steps use features extracted from the output of previous steps. An example of such features is adjacent words from the utterance with filler words and some likely edited words removed.

3 Overview of SALT Corpora

We explore nine corpora included with the SALT software. Table 1 has a high level overview of these corpora, showing where each was collected, the age ranges of the speakers, and the size of each corpus both in terms of transcripts and utterances. Note that only utterances spoken by the child are counted, as we throw out all others.

Table 1 shows several divisions among the corpora. We see that one group of corpora comes from New Zealand, while the majority come from North America. All of the corpora, except for Expository, include children at very different stages of language development.

Four research groups were responsible for the transcriptions and annotations of the corpora in Table 1. One group produced the CONVERSATION, EXPOSITORY, NARRATIVESSS, and NARRATIVESTORYRETELL corpora. Another was responsible for all of the corpora from New Zealand. Finally, the ENNI and GILLAMNT corpora were transcribed and annotated by two different groups. For more details on these corpora, how they were collected, and the annotators, we refer the reader to the SALT website at `http://www.saltsoftware.com/resources/databases.html`.

Some basic inspection reveals that the corpora can be put into three groups based on the median utterance lengths, and the distribution of ut-

Table 1: Description of SALT corpora

Corpus	Transcripts	Utterances	Age Range	Speaker Location
CONVERSATION	584	82,643	2;9 – 13;3	WI & CA
ENNI	377	56,108	3;11 – 10;0	Canada
EXPOSITORY	242	4,918	10;7 – 15;9	WI
GILLAMNT	500	40,102	5;0 – 11;11	USA
NARRATIVESSS	330	16,091	5;2 – 13;3	WI & CA
NARRATIVESTORYRETELL	500	14,834	4;4 – 12;8	WI & CA
NZCONVERSATION	248	25,503	4;5 – 7;7	NZ
NZPERSONALNARRATIVE	248	20,253	4;5 – 7;7	NZ
NZSTORYRETELL	264	2,574	4;0 – 7;7	NZ

terance[2] lengths, following the groups Figure 1, with the EXPOSITORY and CONVERSATION corpora in their own groups. Note that the counts in Figure 1 are of all of the words in each utterance, including those in mazes. We see that the corpora in Group A have a modal utterance length ranging from seven to ten words. There are many utterances in these corpora that are shorter or longer than the median length. Compared to the corpora in Group A, those in Group B have a shorter modal utterance length, and fewer long utterances. In Figure 1, we see that the CONVERSATION corpus consists mostly of very short utterances. At the other extreme is the EXPOSITORY corpus, which resembles the corpora in Group A in terms of modal utterance length, but which generally contains longer utterances than any of the other corpora.

4 Maze Detection Experiments

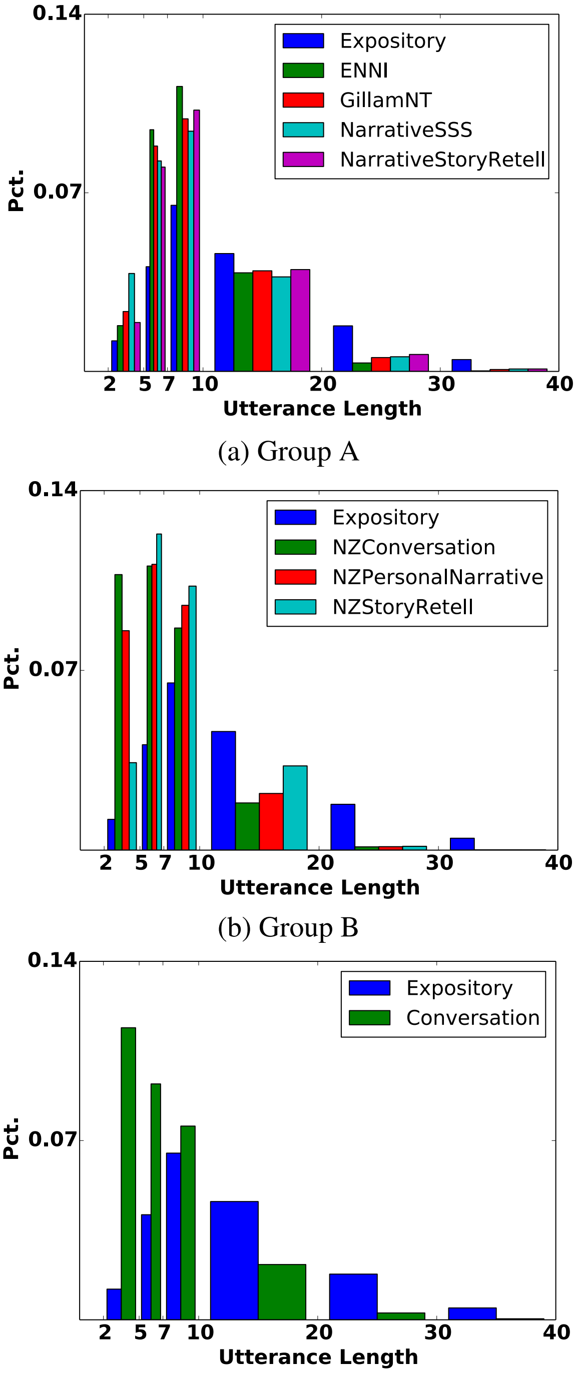
4.1 Maze Detector

We carry out our experiments in automatic maze detection using a statistical maze detector that learns to identify mazes from manually labeled data using features extracted from words and automatically predicted part of speech tags. The maze detector uses the feature set shown in Table 2. This set of features is identical to the ones used by the ‘filler word’ detector in Qian and Liu’s disfluency detector (2013). We also use the same classifier as the second and third steps of their system: the Max Margin Markov Network ‘M3N’ classifier in the pocketcrf toolkit (available at `http://code.google.com/p/pocketcrf/`). The M3N classifier is a kernel-based classifier that is able to leverage the sequential nature the data in this problem (Taskar et al., 2003). We use the following label set: S-O (not in maze); S-M (single word maze); B-M (beginning of multi-word

[2] All of these corpora are reported to have been segmented into *c-units*, which is defined as “an independent clause with its modifiers” (Miller et al., 2011).

Table 2: Feature templates for maze word detection, following Qian and Liu (2013). We extract all of the above features from both words and POS tags, albeit separately. t_0 indicates the current word or POS tag, while t_{-1} is the previous one and t_1 is the following. The function $I(a, b)$ is 1 if a and b are identical, and otherwise 0. y_{-1} is the tag predicted for the previous word.

Category	Features
Unigrams	$t_{-2}, t_{-1}, t_0, t_1, t_2$
Bigrams	$t_{-1}t_0, t_0t_1$
Trigrams	$t_{-2}t_{-1}t_0, t_{-1}t_0t_1, t_0t_1t_2$
Logic Unigrams	$I(t_i, t_0), I(p_i, p_0)$; $-4 \leq i \leq 4; i \neq 0$
Logic Bigrams	$I(t_{i-2}t_{i-1}, t_{-1}t_0)$ $I(t_it_{i+1}, t_0t_{i+1})$; $-4 \leq i \leq 4; i \neq 0$
Predicted tag	y_{-1}



(a) Group A

(b) Group B

(c) Others

Figure 1: Histograms of utterance length (including words in mazes) in SALT corpora

maze); I-M (in multi-word maze); and E-M (end of multi-word maze). The M3N classifier allows us to set a unique penalty for each pair of confused labels, for example penalizing an erroneous prediction of S-O (failing to identify maze words) more heavily than spurious predictions of maze words (all -M labels). This ability is particularly useful for maze detection because maze words are so infrequent compared to words that are not in mazes.

4.2 Evaluation

We split each SALT corpus into training, development, and test partitions. Each training partition contains 80% of the utterances the corpus, while the development and test partitions each contain 10% of the utterances. We use the development portion of each corpus to set the penalty matrix system to roughly balance precision and recall.

We evaluate maze detection in terms of both *tagging* performance and *bracketing* performance, both of which are standard forms of evaluation for various tasks in the Natural Language Processing literature. *Tagging* performance captures how effectively maze detection is done on a word-by-word basis, while *bracketing* performance describes how well each maze is identified in its entirety. For both tagging and bracketing performance, we count the number of true and false positives and negatives, as illustrated in Figure 2. In tagging performance, each word gets counted once, while in bracketing performance we compare the predicted and observed maze spans. We use these counts to compute the following metrics:

$$\text{(P)recision} = \frac{tp}{tp + fp}$$

$$(R)ecall = \frac{tp}{tp + fn}$$

$$\text{F1} = \frac{2PR}{P + R}$$

Note that partial words and punctuation are both ignored in evaluation. We exclude punctuation because punctuation does not need to be included in mazes: it is not counted in summary statistics (e.g. MLU, word count, etc.), and punctuation errors are not captured by the SALT error codes. We exclude partial words because they are always in mazes, and therefore can be detected trivially with a simple rule. Furthermore, because partial words are excluded from evaluation, the performance metrics are comparable across corpora, even if they vary widely in the frequency of partial words.

For both space and clarity, we do not present the complete results of every experiment in this paper, although they are available online[3]. Instead, we present the complete baseline results, and then report F1 scores that are significantly better than the baseline. We establish statistical significance by using a randomized paired-sample test (see Yeh (2000) or Noreen (1989)) to compare the baseline system (system A) and the proposed system (system B). First, we compute the difference d in F1 score between systems A and B. Then, we repeatedly construct a random set of predictions for each input item by choosing between the outputs of system A and B with equal probability. We compute the F1 score of these random predictions, and if it exceeds the F1 score of the baseline system by at least d, we count the iteration as a success. The significance level is at most the number of successes divided by one more than the number of trials (Noreen, 1989).

4.3 Baseline Results

For each corpus, we train the maze detector on the training partition and test it on the development partition. The results of these runs are in Table 3, which also includes the rank of the size of each corpus (1 = biggest, 9 = smallest). We see immediately that our maze detector performs far better on some corpora than on others, both in terms of tagging and bracketing performance. We note that maze detection performance is not solely determined by corpus size: tagging performance is substantially worse on the largest corpus (CONVERSATION) than the small-

Figure 2: Tagging and bracketing evaluation for maze detection. TP = True Positive, FP = False Positive, TN = True Negative, FN = False Negative

Pred.	(	and then it	)	oh		and then it	(	um	)	put his wings out .
Gold	(	and then it		oh	)	and then it	(	um	)	put his wings out .
Tag		TP ×3		FN		TN ×3		TP		TN ×4
Brack.		FP, FN						TP		

[3] http://bit.ly/1dtFTPl

		Tagging			Bracketing		
Corpus	Size Rank	P	R	F1	P	R	F1
CONVERSATION	1	0.821	0.779	0.800	0.716	0.729	0.723
ENNI	2	0.923	0.882	0.902	0.845	0.837	0.841
EXPOSITORY	8	0.703	0.680	0.691	0.620	0.615	0.618
GILLAMNT	3	0.902	0.907	0.904	0.827	0.843	0.835
NARRATIVESSS	6	0.781	0.768	0.774	0.598	0.679	0.636
NARRATIVESTORYRETELL	7	0.799	0.774	0.786	0.627	0.671	0.649
NZCONVERSATION	4	0.832	0.835	0.838	0.707	0.757	0.731
NZPERSONALNARRATIVE	5	0.842	0.835	0.838	0.707	0.757	0.731
NZSTORYRETELL	9	0.905	0.862	0.883	0.773	0.780	0.776

Table 3: Baseline maze detection performance on development sections of SALT corpora: corpus-specific models

est (NZSTORYRETELL).

4.4 Generic Model

We train a generic model for maze detection on all of the training portions of the nine SALT corpora. We use the combined development sections of all of the corpora to tune the loss matrix for balanced precision and recall. We then test the resulting model on the development section of each SALT corpus, and evaluate in terms of tagging and bracketing accuracy.

We find that the generic model performs worse than the baseline in terms of both tagging and bracketing performance on six of the nine corpora corpora. The generic model significantly improves tagging (F1=0.925, $p \leq 0.0022$) on the NZSTORYRETELL corpus, but the improvement in bracketing performance is not significant ($p \leq 0.1635$). There is improvement of both tagging (F1=0.805, $p \leq 0.0001$) and bracketing (F1=0.677, $p \leq 0.0025$) performance on the NARRATIVESSS corpus. The generic model does not perform better than the baseline corpus-specific models on any other corpora.

The poor performance of the generic model is somewhat surprising, as it is trained with far more data than any of the corpus-specific models. In many tasks in natural language processing, increasing the amount of training data improves the resulting model, although this is not necessarily the case if the additional data is noisy or out-of-domain. This suggests two possibilities: 1) the language in the corpora varies substantially, perhaps due to the speakers' ages or the activity that was transcribed; and 2) the maze annotations are inconsistent between corpora.

4.5 Multi-Corpus Models

It is possible that poor performance of the generic model relative to the baseline corpus-specific models can be attributed to systematic differences between the SALT corpora. We may be able to train a model for a set of corpora that share particular characteristics that can outperform the baseline models because such a model could leverage more training data. We first evaluate a model for corpora that contain transcripts collected from children of similar ages. We also evaluate task-specific models, specifically a maze-detection model for story retellings, and another for conversations. These two types of models could perform well if children of similar ages or performing similar tasks produce mazes in a similar manner. Finally, we train models for each group of annotators to see whether systematic variation in annotation standards between research groups could be responsible for the generic model's poor performance.

We train all of these models similarly to the generic model: we pool the training sections of the selected corpora, train the model, then test on the development section of each selected corpus. We use the combined development sections of the selected corpora to tune the penalty matrix to balance precision and recall.

Again, we only report F1 scores that are higher than the baseline model's, and we test whether the improvement is statistically significant. We do not report results where just the precision or just the recall exceeds the baseline model performance, but not F1, because these are typically the result of model imbalance, favoring precision at the expense of recall or vice versa. Bear in mind that we roughly balance precision and recall on the combined development sets, not each corpus's development set individually.

4.5.1 Age-Specific Model

We train a single model on the following corpora: ENNI, GILLAMNT, NARRATIVESSS, and NARRATIVESTORYRETELL. As shown in Table 1, these corpora contain transcripts collected from children roughly aged 4-12. In three of the four corpora, the age-based model performs worse than the baseline. The only exception is NAR-

RATIVESTORYRETELL, for which the age-based model outperforms the baseline in terms of both tagging (F1=0.794, $p \leq 0.0673$) and bracketing (F1=0.679, $p \leq 0.0062$).

4.5.2 Task-Specific Models

We construct two task-specific models for maze detection: one for conversations, and the other for narrative tasks. A conversational model trained on the CONVERSATION and NZCONVERSATION corpora does not improve performance on either corpus relative to the baseline. A model for narrative tasks trained on the ENNI, GILLAMNT, NARRATIVESSS, NARRATIVESTORYRETELL, NZPERSONALNARRATIVE and NZSTORYRETELL corpora only improves performance on one of these, relative to the baseline. Specifically, the narrative task model improves performance on the NARRATIVESSS corpus both in terms of tagging (F1=0.797, $p \leq 0.0005$) and bracketing (F1=0.693, $p \leq 0.0002$).

4.5.3 Research Group-Specific Models

There are two groups of researchers that have annotated multiple corpora: a group in New Zealand, which annotated the NZCONVERSATION, NZPERSONALNARRATIVE, and NZSTORYRETELL corpora; and another group in Wisconsin, which annotated the CONVERSATION, EXPOSITORY, NARRATIVESSS, and NARRATIVESTORYRETELL corpora. We trained research group-specific models, one for each of these groups.

Overall, these models do not improve performance. The New Zealand research group model does not significantly improve performance on any of the corpora they annotated, relative to the baseline. The Wisconsin research group model yields significant improvement on the NARRATIVESSS corpus, both in terms of tagging (F1=0.803, $p \leq 0.0001$) and bracketing (F1=0.699, $p \leq 0.0001$) performance. Performance on the CONVERSATION and EXPOSITORY corpora is lower with the Wisconsin research group model than with the corpus-specific baseline models, while performance on NARRATIVESTORYRETELL is essentially the same with the two models.

5 Discussion

We compared corpus-specific models for maze detection to more generic models applicable to multiple corpora, and found that the generic models performed worse than the corpus-specific ones. This was surprising because the more generic models were able to leverage more training data than the corpus specific ones, and more training data typically improves the performance of data-driven models such as our maze detector. These results strongly suggest that there are substantial differences between the nine SALT corpora.

We suspect there are many areas in which the SALT corpora diverge from one another. One such area may be the nature of the language: perhaps the language differs so much between each of the corpora that it is difficult to learn a model appropriate for one corpus from any of the others. Another potential source of divegence is in transcription, which does not always follow the SALT guidelines (Miller et al., 2011). Two of the idiosyncracies we have observed are: more than three X's (or a consonant followed by multiple X's) to indicate unintelligble language, instead of the conventional X, XX, and XXX for unintelligible words, phrases, and utterances, respectively; and non-canonical transcriptions of what appear to be filled pauses, including ‘uhm’ and ‘umhm’. These idiosyncracies could be straightforward to normalize using automated methods, but doing so requires that they be identified to begin with. Furthermore, although these idiosyncracies may appear to be minor, taken together they may actually be substantial.

Another potential source of variation between corpora is likely in the maze annotations themselves. SALT's definition of mazes, “filled pauses, false starts, and repetitions and revisions of words, morphemes and phrases” (Miller et al., 2011, p. 48), is very short, and none of the components is defined in the SALT manual. In contrast, the Disfluency Annotation Stylebook for Switchboard Corpus (Meteer et al., 1995) describes a system of disfluency annotations over approximately 25 pages, devoting two pages to filled pauses and five to restarts. The Switchboard disfluency annotations are much richer than SALT maze annotations, and we are not suggesting that they are appropriate for a clinical setting. However, between the stark contrast in detail of the two annotation systems' guidelines, and our finding that cross-corpus models for maze detection perform poorly, we recommend that SALT's definition of mazes and their components be elaborated and clarified. This would be of benefit not just to those trying to automate the application of SALT annotations, but also to clinicians who use SALT and depend upon consistently annotated transcripts.

There are two clear tasks for future research that build upon these results. First, maze detection performance can surely be improved. We note, however, that evaluating maze detectors in terms of F1 score may not always be appropriate if such a detector is used in a pipeline. For example, there may be a minimum acceptable level of precision for a maze detector used in a preprocessing step to applying SALT error codes so that maze excision does not create additional errors. In such a scenario, the goal would be to maximize recall at a given level of precision.

The second task suggested by this paper is to explore the hypothesized differences within and between corpora. Such exploration could ultimately result in more rigorous, communicable guidelines for maze annotations, as well as other annotations and conventions in SALT. If there are systematic differences in maze annotations across the SALT corpora, such exploration could suggest ways of making the annotations consistent without completely redoing them.

Acknowledgments

We would like to thank members of the ASD research group at the Center for Spoken Language Understanding at OHSU, for useful input into this study: Jan van Santen, Alison Presmanes Hill, Steven Bedrick, Emily Prud'hommeaux, Kyle Gorman and Masoud Rouhizadeh. This research was supported in part by NIH NIDCD award R01DC012033 and NSF award #0826654. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not reflect the views of the NIH or NSF.

References

Eugene Charniak and Mark Johnson. 2001. Edit detection and parsing for transcribed speech. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–9. Association for Computational Linguistics.

Michelle Dunn, Judith Flax, Martin Sliwinski, and Dorothy Aram. 1996. The use of spontaneous language measures as criteria for identifying children with specific language impairment: An attempt to reconcile clinical and research incongruence. *Journal of Speech and Hearing research*, 39(3):643.

Ling-yu Guo, J Bruce Tomblin, and Vicki Samelson. 2008. Speech disruptions in the narratives of english-speaking children with specific language impairment. *Journal of Speech, Language, and Hearing Research*, 51(3):722–738.

John J Heilmann. 2010. Myths and realities of language sample analysis. *SIG 1 Perspectives on Language Learning and Education*, 17(1):4–8.

Mark Johnson and Eugene Charniak. 2004. A tag-based noisy-channel model of speech repairs. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 33–39, Barcelona, Spain, July.

Judith R Johnston. 2006. *Thinking about child language: Research to practice*. Thinking Publications.

Willem JM Levelt. 1993. *Speaking: From intention to articulation*, volume 1. MIT press, Cambridge, MA.

Barbara G MacLachlan and Robin S Chapman. 1988. Communication breakdowns in normal and language learning-disabled children's conversation and narration. *Journal of Speech and Hearing Disorders*, 53(1):2.

Marie W Meteer, Ann A Taylor, Robert MacIntyre, and Rukmini Iyer. 1995. *Dysfluency annotation stylebook for the switchboard corpus*. University of Pennsylvania.

Jon Miller and Robin Chapman. 1985. Systematic analysis of language transcripts. *Madison, WI: Language Analysis Laboratory*.

Jon F Miller, Karen Andriacchi, and Ann Nockerts. 2011. *Assessing language production using SALT software: A clinician's guide to language sample analysis*. SALT Software, LLC.

Christine Nakatani and Julia Hirschberg. 1993. A speech-first model for repair detection and correction. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 46–53, Columbus, Ohio, USA, June. Association for Computational Linguistics.

Marilyn A Nippold, Tracy C Mansfield, Jesse L Billow, and J Bruce Tomblin. 2008. Expository discourse in adolescents with language impairments: Examining syntactic development. *American Journal of Speech-Language Pathology*, 17(4):356–366.

Eric W Noreen. 1989. Computer intensive methods for testing hypotheses. an introduction. 1989. *John Wiley & Sons*, 2(5):33.

Albert Postma and Herman Kolk. 1993. The covert repair hypothesis: Prearticulatory repair processes in normal and stuttered disfluencies. *Journal of Speech and Hearing Research*, 36(3):472.

Xian Qian and Yang Liu. 2013. Disfluency detection using multi-step stacked learning. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 820–825, Atlanta, Georgia, June. Association for Computational Linguistics.

Christina Reuterskiöld Wagner, Ulrika Nettelbladt, Birgitta Sahlén, and Claes Nilholm. 2000. Conversation versus narration in pre-school children with language impairment. *International Journal of Language & Communication Disorders*, 35(1):83–93.

Matthew Rispoli, Pamela Hadley, and Janet Holt. 2008. Stalls and revisions: A developmental perspective on sentence production. *Journal of Speech, Language, and Hearing Research*, 51(4):953–966.

Cheryl M Scott and Jennifer Windsor. 2000. General language performance measures in spoken and written narrative and expository discourse of school-age children with language learning disabilities. *Journal of Speech, Language & Hearing Research*, 43(2).

Eleanor Messing Semel, Elisabeth Hemmersam Wiig, and Wayne Secord. 2003. *Clinical evaluation of language fundamentals*. The Psychological Corporation, A Harcourt Assessment Company, Toronto, Canada, fourth edition.

Helen Tager-Flusberg, Sally Rogers, Judith Cooper, Rebecca Landa, Catherine Lord, Rhea Paul, Mabel Rice, Carol Stoel-Gammon, Amy Wetherby, and Paul Yoder. 2009. Defining spoken language benchmarks and selecting measures of expressive language development for young children with autism spectrum disorders. *Journal of Speech, Language and Hearing Research*, 52(3):643.

Ben Taskar, Carlos Guestrin, and Daphne Koller. 2003. Maximum-margin markov networks. In *Neural Information Processing Systems (NIPS)*.

Elin T Thordardottir and Susan Ellis Weismer. 2001. Content mazes and filled pauses in narrative language samples of children with specific language impairment. *Brain and cognition*, 48(2-3):587–592.

Danielle Wetherell, Nicola Botting, and Gina Conti-Ramsden. 2007. Narrative in adolescent specific language impairment (sli): A comparison with peers across two different narrative genres. *International Journal of Language & Communication Disorders*, 42(5):583–605.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 947–953. Association for Computational Linguistics.

Simon Zwarts and Mark Johnson. 2011. The impact of language models and loss functions on repair disfluency detection. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 703–711, Portland, Oregon, USA, June. Association for Computational Linguistics.