Using Constituency and Dependency Parse Features to Identify Errorful Words in Disordered Language

Eric Morley and Emily Prud'hommeaux
Center for Spoken Language Understanding
Oregon Health & Science University
Portland, OR, USA
{morleye, emilypx}@gmail.com

Abstract

Delayed or disordered language is a characteristic of both autism spectrum disorder (ASD) and specific language impairment (SLI). In this paper, we describe our data set, which consists of transcribed data from a widely used clinical diagnostic instrument (the ADOS) for children with ASD and children with SLI. These transcripts are manually annotated with SALT, an annotation system that applies a descriptive code to errorful words. Here we address a step in automating SALT annotation: identifying the errorful words in sentences that are known to contain an error. We propose a set of baseline features to identify errorful words, and investigate the effectiveness of adding features extracted from dependency and constituency parses. We find that features from both types of parses improve classifier performance above our baseline, both individually and in aggregate.

1. Introduction

The language of children with neurological disorders affecting communication is characterized by a disproportionate number of morphological and syntactic errors relative to the language of age-matched peers [1, 2]. These include not only the kinds of errors that are observed during typical language acquisition, such as overregularization, but also idiosyncratic errors in word order and agreement that cannot be explained by delayed acquisition of morphological or syntactic rules. The patterns and distributions of these two types of errors can indicate the presence of autism spectrum disorder, an intellectual disability, or specific language impairment [2, 3].

Highly structured language assessment instruments, such as the Clinical Evaluation of Language Fundamentals [4], may be unable to elicit these types of diagnostically informative errors. Analysis of natural language samples is often able to reveal a wider variety of errors than structured instruments, but the reliable manual annotation of errors in spoken language transcripts in accordance with the coding guidelines of IPSyn [5] and the Systematic Analysis of Language Transcripts (SALT) [6] or other standard analysis instruments, requires significant expertise, time, and resources.

In this paper, we present work in using part-of-speech and parse features to automatically locate syntactic and morphological errors in spontaneous language samples of children with autism spectrum disorder (ASD) and specific language impairment (SLI). The system described here is intended to be one step in a pipeline that identifies sentences containing an error, determines which words are the source of the error, and thus should be marked as *errorful*, and classifies each of the errors. Here we address only the second step: identifying errorful words in sentences that are known to contain an error. Despite having only a small number of training examples, we are able to locate errorful words with high accuracy, demonstrating the potential of natural language processing and machine learning techniques for the task of analyzing spontaneous child language samples.

2. Background

Volden and Lord [2] were among the first to analyze the specific types of syntactic errors produced by children with ASD. They found that compared to cognitively matched children with typical development, children with ASD generated significantly more syntactic errors that could not be explained by the delayed acquisition of a known syntactic or morphological rule (e.g., *But in the car, it's some.*). The language of children with an intellectual disability, in contrast, was characterized by an increased number of developmental errors that a child acquiring his first language might be expected to make, such as over regularization (e.g., *he goed*) or incorrect subject-verb agreement (e.g., *What does cows do?*). Similar results were replicated both manually and automatically by Prud'hommeaux et al. in children with ASD and SLI [3]. The work presented here differs from the above in that we will be identifying the word that is the source of the error rather than the presence or absence of a particular error type in a given sentence.

Previous work on automatic annotation of spoken lan-

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.

guage transcripts of children has focused not on identifying and classifying errors, but on assessing the language acquisition progress of children with typical development and language and speech disorders. Long and Channell used existing grammatical tagging software to extract specific syntactic constructions and measures of grammatical complexity used in a variety of language analysis tools [7]. Later work relied on dependency parses to identifying the various syntactic constructions that are counted by the IPSyn language analysis system [8]. Although this research did not address the issue of error detection in disordered language, it did demonstrate that existing tagging and parsing tools designed primarily for the written language of adults could be adapted to child language analysis.

3. Data

In this investigation, we use data from children with a diagnosis of ASD ($n = 34$) or SLI ($n = 17$). No typically developing children participated in this study. Subjects ranged in aged between 4 and 8 years and were required to be intelligible, to have a full-scale IQ of greater than 70, and to have a mean length of utterance (MLU) of at least 3. A diagnosis of ASD was indicated if a child exceeded the cut-off for two distinct diagnostic instruments and met the diagnostic criteria outlined in the DSM-IV [9]. For this particular study, a child received a diagnosis of SLI if he met one of two commonly used criteria: 1) The Tomblin Epi-SLI criteria [10], in which diagnosis of language impairment is indicated when scores in two out of five domains (vocabulary, grammar, narrative, receptive language, and expressive language) are greater than 1.25 standard deviations below the mean; or 2) the CELF [4] criteria, in which diagnosis of language impairment is indicated when one out of three index scores and one out of three spontaneous language scores are more than one standard deviation below the mean.

Our corpus consists of transcripts of the Autism Diagnostic Observation Schedule (ADOS) [11]. The ADOS is a diagnostic instrument consisting of a semi-structured series of activities designed to elicit behaviors associated with ASD. The transcripts of the study participants were manually annotated by trained speech language pathologists according to the coding guidelines for the SALT.

All errors in the transcripts have been manually annotated with SALT error labels. The exact set of error codes used in SALT varies according to the particular needs of the coder, although there are some generally agreed upon codes that are used by most clinicians. In the coding scheme used by our speech language pathologists, the available word-level error labels include, but are not limited to, the following, with examples of each error code given in parentheses: [EO] - overgeneralization (*He fell/ed[EO] down*); [EW] - generic word error (*They came to a stop/ed[EW]*); [WO] - unconventional word order (*I don't know what is it[WO]*); and [EX] - extraneous words (*The boy is an[EX] sleeping*). Note that some error tags may apply to grammatical words and sentences due to semantic considerations. For example, in one case, "his" in *My sister ate his[EW] dinner* is corrected to "her".

These transcripts contain 20,314 sentences, of which 1,150 contain an error. Within the 1,150 sentences that contain an error, there are 6,306 words, 1,268 of which are marked as containing an error[1]. In this investigation, we only use these 6,306 words from sentences known to contain at least one error, and the set of error tags for each word is collapsed into *errorful* (the word has/is preceded by an error) or not.

4. Features and Classification

We consider three sets of features to classify each word as errorful or not: 1) baseline, 2) dependency parse, and 3) constitutent parse features. These features, as well as classification, are discussed below.

4.1. Baseline Features

We use the following features as our baseline:

1. INWSJ is true for word w if w appears in the Penn Treebank Wall Street Journal (WSJ) corpus. (Boolean feature)
2. P-POS-BIGRAM-CHILDES is the probability of observing the part-of-speech (POS) bigram starting at word w in the CHILDES corpus [12]. This feature is missing for the last word of each sentence because there is no POS bigram starting at that word. (Numeric feature)
3. P-POS-TRIGRAM-CHILDES is the probability of observing the part-of-speech (POS) trigram starting at word w in the CHILDES corpus. This feature is missing for the last two words in each sentence. (Numeric feature)
4. P-POS-BIGRAM-WSJ is identical to P-POS-BIGRAM-CHILDES, except that the probability is taken from the WSJ rather than CHILDES. (Numeric feature)
5. P-POS-TRIGRAM-WSJ is identical to P-POS-TRIGRAM-CHILDES, except that the probability is taken from the WSJ rather than CHILDES. (Numeric feature)

The INWSJ feature is intended to capture non-standard words, for example ungrammatical forms such

[1]For tags indicating omitted words or morphemes, we tag the word following the omitted item as being preceded by an omitted word or morpheme, as appropriate.

as GOED, and neologisms. For INWSJ, we chose to use the WSJ instead of CHILDES because the WSJ should contain fewer errorful words. For example **goed* does not appear in the WSJ, but it does in CHILDES.

We use the hand-annotated POS tags for the features based on POS n-grams in the WSJ. For the ones based on POS n-grams in the CHILDES, we use POS tags produced by TnT [13]. These features should capture some unusual constructions, for example, an article followed immediately by a verb, as would happen if the subject of the sentence were omitted.

4.2. Dependency Parse Features

We parse each sentence with the Stanford Dependency Parser [14], then extract the following features from the dependency parses (some with reference to dependency parses of the CHILDES corpus). All of these features are boolean features, unless noted otherwise. We include an intuitive description of each feature as the last item in the description. Note, however, that these features are extracted automatically, and there may be some exceptions to the intuitive descriptions (for example, the *nsubj* relationship in the first feature may be between a noun and an adjective).

1. MISMATCHED-NUMBER is true for any words in a dependency relationship labeled *nsubj* or *det* in which one word is singular and the other is plural, and otherwise false. This feature is true if the subject of a sentence is singular, but the verb is plural (for example "John go to the store.")

2. HAS-NSUBJ is true for any word in a sentence that contains a dependency arc labeled *nsubj*, and is otherwise false. This feature is true if there is a subject and main verb in the sentence.

3. NSUBJ-ACC is true for oblique pronouns (me, him her, us, them), if they appear in a dependency relationship labeled *nsubj*. In other words, this feature is true if the subject of the sentence is an oblique pronoun.

4. NON-FINITE-MAIN-VB is true for any word in a dependency relationship labeled *nsubj* if neither word in that relationship is a finite verb. This feature is true if the main verb in the sentence is a participle.

5. PROB-DEP-CHILDES is the probability of observing word w on the same side of an an arc labeled l in the CHILDES corpus (also parsed with the Stanford parser). For example, if we have the dependency arc `nn(engine-9, fire-8)`, then the feature for the word *fire is the probability of observing* fire on the right side of an arc labeled `nn` in the CHILDES corpus. This feature captures how likely the observed dependency relationship is in a corpus of speech collected from typically developing children. (Numeric feature)

4.3. Constituency Parse Features

For constituency parse features, we simply use the word-level features extracted by Roark's incremental top-down parser [15], which was trained on the Switchboard corpus. This parser uses a beam-search, and this beam has an impact on the features below. The word-level features are all numeric features. We describe each feature at word w_i, but they are discussed in far more detail in a technical report by Roark [16]. These descriptions are somewhat technical, but we have attempted to make them as intuitive as possible.

1. PREFIX - is the sum of probabilities of observing all possible trees spanning words w_1 throughw_i in which the last rule applied is generating the terminal w_i on the right hand side.

2. SRPRSL - is the surprisal at w_i, and it is the sum of two other features: LEXSRPRSL and SYNSRPRSL. As Roark et al. note, "high surprisal scores result when the prex probability at w_i is low relative to the prex probability at w_{i-1}" [15].

3. LEXSRPRSL - is the surprisal at w_i due to the identity of w_i.

4. SYNSRPRSL - is the surprisal at w_i due to the syntactic structure that integrating w_i would create.

5. AMBIG - is the ambiguity of w_i, measured as the entropy over the beam used in the beam search.

6. OPEN - is the weighted average number of open brackets in the beam.

7. RERNK - is the ratio of the probability of parses extending the top-ranked parse at w_i to the end of the sentence to the probability of the top-ranked parse at w_i. These two probabilities are normalized over the beam, so RERNK can be greater than 1.

8. TOPERR - is the ratio of the conditional probability of the top-ranked parse at w_{i+1} to the one at w_i. If the parse at w_{i+1} is not an extension of the top ranked parse at w_i, TOPERR is 0.

9. STPS - is the weighted average number of steps in the derivation in the beam from w_{i-1} to w_i.

4.4. Classification

We classify each word in the corpus as being errorful or not using the Random Forest classifier with default settings in the Weka machine learning toolkit [17]. This classifier is fast to train and avoids overfitting [18]. In

Features	P	R	F1	AUC
1	0.532	0.230	0.321	0.581
1-3	0.542	0.244	0.336	0.697
1, 4-5	**0.567**	0.208	0.304	0.675
1-5	0.545	**0.308**	**0.393**	**0.723**

Table 1: Classification performance with combinations of baseline features

informal tests, we found that this classifier yielded the highest performance of any in Weka. For these reasons, we chose to use it for further experiments. We report results from 10-fold cross evaluation. Due to the high percentage (79.9%) of error-free words, we report the precision, recall and f-measure of identifying errorful words. In addition, we report the area under the receiver operating characteristic curve (AUC), which ranges from 0.5 for a classifier performing at chance to 1.0 for a perfect classification [19].

5. Experiments and Results

First we examine the effectiveness of various combinations of baseline features. We report the precision and recall of detecting errorful words. The results are reported in Table 5. We will use all five baseline features in all further trials because doing so yielded the highest f-measure of all the combinations of baseline features that we tested. Note that the precision, recall and f-measure we report is for identifying errorful words, not for correct classification.

Next, we investigate the effects of adding features extracted from dependency and constituency parses to the baseline features. We report the performance yielded by adding each of the dependency-parse features to the baseline individually in Table 5. All of the features except for HAS-NSUBJ and NON-FINITE-MAIN-VB improve classification. We also report classification performance with: 1) all of the dependency parse features (DEPALL); 2) DEP-ALL with HAS-NSUBJ omitted (DEPALL-NONSUBJ); 3) DEP-ALL with NON-FINITE-MAIN-VB omitted (DEPALL-NONFMV); and 4) DEP-ALL with both HAS-NSUBJ and NON-FINITE-MAIN-VB omitted (DEP-IMPRONLY) in 5.

We report the performance yielded by adding each of the constituency parse features to the baseline individually in Table 5. All of the features except for HAS-NSUBJ and NON-FINITE-MAIN-VB improve classification. We also report classification performance with: 1) all of the dependency parse features (DEPALL); 2) DEP-ALL with HAS-NSUBJ omitted (DEPALL-NONSUBJ); 3) DEP-ALL with NON-FINITE-MAIN-VB omitted (DEPALL-NONFMV); and 4) DEP-ALL with both HAS-NSUBJ and NON-FINITE-MAIN-VB omitted (DEP-IMPRONLY) in 5.

Features	P	R	F1	AUC
MISMATCHED-NUMBER	0.544	0.310	0.395	0.727
HAS-NSUBJ	0.527	0.311	0.391	0.723
NSUBJ-ACC	0.554	0.312	0.399	0.733
NON-FINITE-MAIN-VB	0.530	0.311	0.392	0.721
PROB-DEP-CHILDES	**0.566**	**0.349**	**0.432**	**0.759**
DEPALL	0.561	0.348	0.429	**0.765**
DEPALL-NONSUBJ	**0.580**	**0.360**	**0.445**	0.760
DEPALL-NONFMV	0.554	0.341	0.423	0.762
DEP-IMPRONLY	0.569	0.350	0.434	**0.765**

Table 2: Classification performance with baseline and dependency-parse features

Features	P	R	F1	AUC
PREFIX	0.531	0.345	0.418	0.721
SRPRSL	0.542	0.378	0.445	0.739
SYNSP	0.513	0.352	0.417	0.724
LEXSP	**0.546**	**0.380**	**0.448**	**0.743**
AMBIG	0.512	0.338	0.407	0.714
OPEN	0.527	0.341	0.414	0.721
RERNK	0.502	0.321	0.392	0.709
TOPERR	0.520	0.335	0.408	0.707
STPS	0.531	0.316	0.396	0.716
CNSALL	**0.601**	**0.371**	**0.459**	**0.767**
CNSALL-NORENRK	0.590	0.370	0.455	0.766

Table 3: Classification performance with baseline and constituency parse features

Finally, we combine the most successful feature-sets found above. The results are reported in Table 5. With one exception (DEPALL + CNSALL), classification performance for all combined feature sets exceeds the performance of their component sets. The two best performing feature sets are DEPALL-NONSUBJ+CNSALL and DEPALL-NONSUBJ+CNSALL-NORERNK. It is not surprising that DEPALL-NONSUBJ yields better classification performance than DEPALL including NSUBJ given the results in Table 5. It is surprising, however, that removing RERNK from CNSALL should improve classification performance when dependency parse features are included, since CNSALL outperformed CNSALL-NORERNK (see Table 5).

6. Conclusions and Future Directions

We have proposed three sets of features for identifying errorful words in an utterance that is known to contain at least one such word. The baseline features are all based on the words themselves and POS tags, and are therefore quite easy to extract. In addition to these baseline features, we propose five features extracted from dependency parses and nine features extracted from constituency parses. Three of the five dependency parse features improved performance when added to the baseline individually. However, we found that adding four of these features in tandem yielded higher classification performance than adding either the three ones that improved

Features	P	R	F1	AUC
DEPALL + CNSALL	0.618	0.365	0.459	0.770
DEPALL + CNSALL-NORENRK	0.620	0.375	0.467	**0.782**
DEPALL-NONSUBJ + CNSALL	0.617	0.370	0.463	0.771
DEPALL-NONSUBJ + CNSALL-NORERNK	**0.642**	**0.378**	**0.476**	0.779
DEP-IMPRONLY + CNSALL	0.622	0.369	0.463	0.768
DEP-IMPRONLY + CNSALL-NORERNK	0.617	0.375	0.467	0.777

Table 4: Classification performance with baseline, dependency and constituency parse features

performance independently, or all of the dependency-parse features. This suggests that observing various combinations of certain dependency-parse features can yield more information than observing them separately. It also suggests that certain dependency parse features, for example NON-FINITE-MAIN-VB, were often triggered erroneously, thus making them uninformative at best. Of the nine features extracted from the top-down incremental constituency parser, eight yielded improved classification performance. Incorporating all of the constituency parser increased performance more than any feature alone, and also more than the eight features that improved performance independently.

Combining the baseline features with the best sets of dependency and constituency parse features yielded the highest performance we observed. Relative to the baseline, we improved precision by 17.9%, recall by 22.7%, f-measure by 21.1%, and AUC by 8.1%. We expect that extracting other, possibly more sophisticated, features from parses will yield higher performance in detecting errorful words. Although there has been, to our knowledge, no previous work in automated word-level error detection in the language of children with developmental disorders, we note that these results compare favorably with results reported in error identification in the writing of second language learners [20, 21].

In future work, we plan to implement the two missing elements of the pipeline: identifying sentences that contain errors and classifying the identified errors according to their error code. We expect that many of the features discussed here will be helpful in these two tasks. Surprisal features, for instance, may be helpful in determining whether a sentence contains an error, while many of the dependency features will aid in determining the correct error code. We also plan to explore methods of syntactic error detection used in automated essay scoring. Given the distinctive patterns in language errors produced by children with ASD and language impairments, a complete error detection pipeline has the potential to provide important diagnostic features that could be used along with other information in an automated screening tool for developmental disorders.

7. References

[1] J. B. Tomblin, N. L. Records, P. Buckwalter, X. Z. abd Elaine Smith, and M. O'Brien, "Prevalence of specific language impairment in kindergarten children," *Journal of Speech Language and Hearing Research*, vol. 40, pp. 1245–1260, 1997.

[2] J. Volden and C. Lord, "Neologisms and idiosyncratic language in autistic speakers," *Journal of Autism and Developmental Disorders*, vol. 21, pp. 109–130, 1991.

[3] E. Prud'hommeaux, B. Roark, L. Black, and J. van Santen, "Classification of atypical language in autism," in *Proceedings of the 2nd ACL Workshop on Cognitive Modeling and Computational Linguistics (CMCL)*, 2011.

[4] T. Paslawski, "The clinical evaluation of language fundamentals, fourth edition (celf-4): A review," *Canadian Journal of School Psychology*, vol. 20, no. 1-2, pp. 129–134, 2005.

[5] H. S. Scarborough, "Index of productive syntax," *Applied Psycholinguistics*, vol. 11, pp. 1–22, 1990.

[6] J. F. Miller, K. Andriacchi, and A. Nockerts, 2011.

[7] S. Long and R. Channell, "Accuracy of four language analysis procedures performed automatically," *American Journal of Speech-Language Pathology*, vol. 10, no. 2, p. 180, 2001.

[8] K. Sagae, A. Lavie, and B. MacWhinney, "Automatic measurement of syntactic development in child language," in *Proceedings of ACL*, 2005, pp. 197–204.

[9] American Psychiatric Association, *DSM-IV-TR: Diagnostic and Statistical Manual of Mental Disorders.*, 4th ed. Washington, DC: American Psychiatric Publishing, 2000.

[10] J. B. Tomblin, "Genetic and environmental contributions to the risk for specific language impairment," in *Toward a genetics of language*. Lawrence Erlbaum Associates, 1996, pp. 191–211.

[11] C. Lord, M. Rutter, S. Goode, J. Heemsbergen, H. Jordan, L. Mawhood, and E. Schopler, "Austism diagnostic observation schedule: A standardized observation of communicative and social behavior," *Journal of autism and developmental disorders*, vol. 19, no. 2, pp. 185–212, 1989.

[12] B. MacWhinney, *The CHILDES Project: Tools for Analyzing Talk*. Psychology Press, 2000, vol. 2.

[13] T. Brants, "TnT: A statistical part-of-speech tagger," in *Proceedings of the Sixth Conference on Applied Natural Language Processing*. Association for Computational Linguistics, 2000, pp. 224–231.

[14] M. De Marneffe, B. MacCartney, and C. Manning, "Generating typed dependency parses from phrase structure parses," in *Proceedings of LREC*, vol. 6, 2006, pp. 449–454.

[15] B. Roark, A. Bachrach, C. Cardenas, and C. Pallier, "Deriving lexical and syntactic expectation-based measures for psycholinguistic modeling via incremental top-down parsing," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1*. Association for Computational Linguistics, 2009, pp. 324–333.

[16] B. Roark, "Expected surprisal and entropy," Oregon Health & Science University, Tech. Rep., 2011.

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, "The weka data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[18] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[19] J. Egan, *Signal Detection Theory and ROC Analysis*. Academic Press, 1975.

[20] M. Chodorow, J. R. Tetreault, and N.-R. Han, "Detection of grammatical errors involving prepositions," in *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions*, 2007, pp. 25–30.

[21] M. Gamon, J. Gao, C. Brockett, A. Klementiev, W. Dolan, D. Belenko, and L. Vanderwende, "Using contextual speller techniques and language modeling for esl error correction," *Proceedings of*

IJCNLP, 2008.