Error Detection Using Linguistic Features

Yongmei Shi
Department of Computer Science and
Electrical Engineering
University of Maryland Baltimore County
Baltimore, MD 21250
yshi1@umbc.edu

Lina Zhou
Information Systems Department
University of Maryland Baltimore County
Baltimore, MD 21250
zhoul@umbc.edu

Abstract

Recognition errors hinder the proliferation of speech recognition (SR) systems. Based on the observation that recognition errors may result in ungrammatical sentences, especially in dictation application where an acceptable level of accuracy of generated documents is indispensable, we propose to incorporate two kinds of linguistic features into error detection: lexical features of words, and syntactic features from a robust lexicalized parser. Transformation-based learning is chosen to predict recognition errors by integrating word confidence scores with linguistic features. The experimental results on a dictation data corpus show that linguistic features alone are not as useful as word confidence scores in detecting errors. However, linguistic features provide complementary information when combined with word confidence scores, which collectively reduce the classification error rate by 12.30% and improve the F measure by 53.62%.

1 Introduction

The proliferation of speech recognition (SR) systems is hampered by the ever-presence of recognition errors and the significant amount of effort involved in error correction. A user study (Sears et al., 2001) showed that users spent one-third of their time finding and locating errors and another one-third of the time correcting errors in a hand-free dictation task. Successfully detecting SR errors can speed up the entire process of error correction. Therefore, we focus on error detection in this study.

A common approach to detecting SR errors is annotating confidence at the word level. The majority of confidence annotation methods are based on feature combination, which follows two steps: (i) extract useful features characteristics of the correctness of words either from the inner components of an SR system (SR-dependent features) or from the recognition output (SR-independent features); and (ii) develop a binary classifier to separate words into two groups: correct recognitions and errors.

Various features extracted from different components of an SR system, such as the acoustic model, the language model, and the decoder, have been proven useful to detecting recognition errors (Chase, 1997; Pao et al., 1998; San-Segundo et al., 2001). Nonetheless, merely using these features is inadequate, because the information conveyed by these features has already been considered when SR systems generate the output. A common observation is that the combination of SR-dependent features can only marginally improve the performance achieved by using only the best single feature (Zhang and Rudnicky, 2001; Sarikaya et al., 2003). Hence information sources beyond the SR system are desired in error detection.

High-level linguistic knowledge is a good candidate for additional information sources. It can be extracted from the SR output via natural language processing, which compensates for the lack of high-

Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 41–48, Vancouver, October 2005.

level linguistic knowledge in a typical SR system. A user study (Brill et al., 1998) showed that humans can utilize linguistic knowledge at various levels to improve the SR output by selecting the best utterance hypotheses from N-best lists. Linguistic features from syntactic, semantic, and dialogue discourse analyses have proven their values in error detection in domain specific spoken dialogue systems, e.g. (Rayner et al., 1994; Carpenter et al., 2001; Sarikaya et al., 2003). However, few studies have investigated the merit of linguistic knowledge for error detection in dictation, a domain-independent application.

Transformation-based learning (TBL) is a rule-based learning method. It has been used in error correction (Mangu and Padmanabhan, 2001) and error detection (Skantze and Edlund, 2004). The rules learned by TBL show good interpretability as well as good performance. Although statistical learning methods have been widely used in confidence annotation (Carpenter et al., 2001; Pao et al., 1998; Chase, 1997), their results are difficult to interpret. Therefore, we select TBL to derive error patterns from the SR output in this study.

The rest of the paper is organized as follows. In Section 2, we review the extant work on utilizing linguistic features in error detection. In Section 3, we introduce linguistic features used in this study. In Section 4, we describe transformation-based learning and define the transformations, followed with reporting the experimental results in Section 5. Finally, we summarize the findings of this study and suggest directions for further research in Section 6.

2 Related Work

When the output of an SR system is processed, the entire utterance is available and thus utterance-level contextual information can be utilized. Features generated from high-level language processing such as syntactic and semantic analyses may complement the low-level language knowledge (usually n-gram) used in the SR systems.

Most of the previous work on utilizing linguistic features in error detection focused on utterance-level confidence measures. Most of features were extracted from the output of syntactic or semantic parsers, including full/robust/no parse, number of words parsed, gap number, slot number, grammar rule used, and so on (Rayner et al., 1994; Pao et al., 1998; Carpenter et al., 2001; San-Segundo et al., 2001). Some discourse-level features were also employed in spoken dialogue systems such as number of turns, and dialog state (Carpenter et al., 2001).

Several studies incorporated linguistic features into word-level confidence measures. Zhang and Rudnicky (2001) selected two features, i.e., parsing mode and slot backoff mode, extracted from the parsing result of Phoenix, a semantic parser. The above two features were combined with several SR-dependent features using SVM, which achieved a 7.6% relative classification error rate reduction over SR-dependent features on the data from CMU Communicator system.

Sarikaya et al. (2003) explored two sets of semantic features: one set from a statistical classer/parser, and the other set from a maximum entropy based semantic-structured language model. When combined with the posterior probability using the decision tree, both sets achieved about 13-14% absolute improvement on correct acceptance at 5% false acceptance over the baseline posterior probability on the data from IBM Communicator system.

Skantze and Edlund (2004) focused on lexical features (e.g., part-of-speech, syllables, and content words) and dialogue discourse features (e.g., previous dialogue act, and mentioned word), but did not consider parser-based features. They employed transformation-based learning and instance-based learning as classifiers. When combined with confidence scores, the linguistic features achieved 7.8% absolute improvement in classification accuracy over confidence scores on one of their dialogue corpora.

It is shown from the related work that linguistic features have merit in judging the correctness of words and/or utterances. However, such features have only been discussed in the context of conversational dialogue in specific domains such as ATIS (Rayner et al., 1994), JUPITER (Pao et al., 1998), and Communicator (Carpenter et al., 2001; San-Segundo et al., 2001; Zhang and Rudnicky, 2001; Sarikaya et al., 2003).

In an early study, we investigated the usefulness of linguistic features in detecting word errors in dictation recognition (Zhou et al., 2005). The linguistic features were extracted from the parsing result of the link grammar. The combination of linguistic features with various confidence score based features using SVM can improve F measure for error detection from 42.2% to 55.3%, and classification accuracy from 80.91% to 83.53%. However, parser-based features used were limited to the number of links that a word has.

3 Linguistic Features

For each output word, two sets of linguistic features are extracted: lexical features and syntactic features.

3.1 Lexical Features

For each word w, the following lexical features are extracted:

- word: w itself
- pos: part-of-speech tag from Brill's tagger (Brill, 1995)
- syllables: number of syllables in w, estimated based on the distribution patterns of vowels and consonants
- position: the position of w in the sentence: beginning, end, and middle

3.2 Syntactic Features

Speech recognition errors may result in ungrammatical sentences under the assumption that the speaker follows grammar rules while speaking. Such an assumption holds true especially for dictation application because the general purpose of dictation is to create understandable documents for communication.

Syntactic parsers are considered as the closest approximation to this intuition since there is still a lack of semantic parsers for the general domain. Moreover, robust parsers are preferred so that an error in a recognized sentence does not lead to failure in parsing the entire sentence. Furthermore, lexicalized parsers are desired to support error detection at the word level. As a result, we select *Link Grammar*[1] to generate syntactic features.

3.2.1 Link Grammar

Link Grammar is a context-free lexicalized grammar without explicit constituents (Sleator and Temperley, 1993). In link grammar, rules are expressed as link requirements associated with words. A link requirement is a set of disjuncts, each of which represents a possible usage of the word. A sequence of words belongs to the grammar if the result linkage is a planar, connected graph in which at most one link is between each word pair and no cross link exists. Link grammar supports robust parsing by incorporating null links (Grinberg et al., 1995).

3.2.2 Features from Link Grammar

We hypothesize that a word without any link in a linkage of the sentence is a good indicator of the occurrence of errors. Either the word itself or words around it are likely to be erroneous. It has been shown that null links can successfully ignore false starts and connect grammatical phrases in ungrammatical utterances, which are randomly selected from the Switchboard corpus (Grinberg et al., 1995).

A word with links may still be an error, and its correctness may affect the correctness of words linked to it, especially those words connected with the shortest links that indicate the closest connections.

Accordingly, for each word w, the following features are extracted from the parsing result:

- haslink: whether w has left links, right links, or no link
- llinkto/rlinkto: the word to which w links via the shortest left/right link

An example of parsing results is illustrated in Figure 1. Links are represented with dotted lines which are annotated with labels (e.g., Wd, Xp) representing link types. In Figure 1, word "since" has no link, and word "around" has one left link and one right link. The word that has the shortest left link to "world" is "the".

[1] Available via http://www.link.cs.cmu.edu/link/

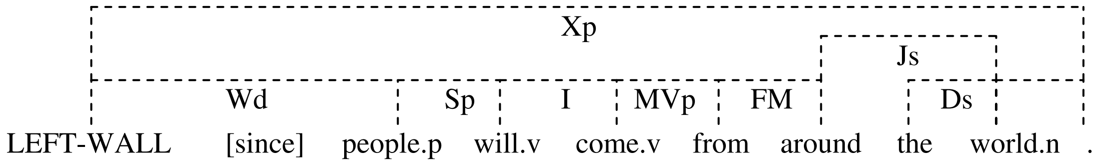

Figure 1: An Example of Parsing Results of Link Grammar

4 Error Detection based on Transformation-Based Learning

4.1 Transformation-Based Learning

Transformation-Based Learning is a rule-based approach, in which rules are automatically learned from the data corpus. It has been successfully used in many natural language applications such as part-of-speech tagging (Brill, 1995). Three prerequisites for using TBL are: an initial state annotator, a set of possible transformations, and an objective function for choosing the best transformations.

Before learning, the initial state annotator adds labels to the training data. The learning goes through the following steps iteratively until no improvement can be achieved: (i) try each possible transformation on the training data, (ii) score each transformation with the objective function and choose the one with the highest score, and (iii) apply the selected transformation to update the training data and append it to the learned transformation list.

4.2 Error Detection Based on TBL

Pre-defined transformation templates are the rules allowed to be used, which play a vital role in TBL. The transformation templates are defined in the following format:

Change the word label of a word w from X to Y, if condition C is satisfied

where, X and Y take binary values: 1 (correct recognition) and -1 (error). Each condition C is the conjunction of sub-conditions in form of $f \ op \ v$, where f represents a feature, v is a possible categorical value of f, and op is the possible operations such as $<$, $>$ and $=$.

In addition to the linguistic features introduced in Section 3, two other features are used:

- word confidence score (CS): an SR dependent feature generated by an SR system.
- word label (label): the target of the transformation rules. Using it as a feature enables the propagation of the effect of preceding rules.

As shown in Table 1, conditions are classified into three categories based on the incrementally enlarged context from which features are extracted: *word alone*, *local context*, and *sentence context*. The three categories are further split into seven groups according to the features they used.

- **L**: the correctness of w depends solely on itself. Conditions only include lexical features of w.
- **Local**: the correctness of w depends not only on itself but also on its surrounding words. Conditions incorporate lexical features of surrounding words as well as those of w. Furthermore, word labels of surrounding words are also employed as a feature to capture the effect of the correctness of surrounding words of w.
- **Long**: the scope of conditions for the correctness of w is expanded to include syntactic features. Syntactic features of w and its surrounding words as well as the features in Local are incorporated into conditions. In addition, the lexical features and word labels of words that have the shortest links to w are also incorporated.
- **CS**: the group in which conditions only include confidence scores of w.
- **LCS, CSLocal, CSLong**: these three groups are generated by combining the features from L, Local, and Long with the confidence scores of w as an additional feature respectively.

$lrHaslink$ and $llinkLabel$ are combinations of basic features. $lrHaslink$ represents whether the preceding word and the following word have links,

Category	Group	Example
Word Alone	CS	$cs(w_i) < c_i$
	L	$position(w_i) = t_i$ & $syllables(w_i) = s_i$
	LCS	$cs(w_i) < c_i$ & $pos(w_i) = p_i$
Local Context	Local	$position(w_i) = t_i$ & $label(w_{i-1}) = l_{i-1}$ & $word(w_i) = d_i$
	CSLocal	$cs(w_i) < c_i$ & $position(w_i) = t_i$ & $label(w_{i-1}) = l_{i-1}$ & $label(w_{i+1}) = l_{i+1}$
Sentence Context	Long	$position(w_i) = t_i$ & $lrHaslink(w_i) = h_i$ & $haslink(w_i) = hl_i$
	CSLong	$cs(w_i) < c_i$ & $position(w_i) = t_i$ & $llinkLabel(w_i) = ll_i$ & $pos(w_i) = p_i$

Table 1: Condition Categories and Examples

and $llinkLabel$ represents the label of the word to which w has the shortest left link. c_i, t_i, s_i, p_i, l_i, d_i, h_i, hl_i, and ll_i are possible values of the corresponding features.

The initial state annotator initializes all the words as correct words. A Prolog based TBL tool, μ-TBL (Lager, 1999) [2] is used in this study. Classification accuracy is adopted as the objective function. For each transformation, its *positive effect* (*PE*) is the number of words whose labels are correctly updated by applying it, and its *negative effect* (*NE*) is the number of words wrongly updated. Two cut-off thresholds are used to select transformations with strong positive effects: net positive effect ($PE - NE$), and the ratio of positive effect ($PE/(PE + NE)$).

5 Experimental Results and Discussion

Experiments were conducted at several levels. Starting with transformation rules with word alone conditions, additional rules with local context and sentence context conditions were incorporated incrementally by enlarging the scope of the context. As such, the results help us not only identify the additional contribution of each condition group to the task of error detection but also reveal the importance of enriching contextual information to error detection.

5.1 Data Corpus

The data corpus was collected from a user study on a composition dictation task (Feng et al., 2003). A total of 12 participants were native speakers and none of them used their voice for professional purposes. Participants spoke to IBM ViaVoice (Millennium edition), which contains a general vocabulary of 64,000 words. The dictation task was completed in a quiet lab environment with high quality microphones.

During the study, participants were given one predesigned topic and instructed to compose a document of around 400 words on that topic. Before starting the dictation, they completed enrollments to build personal profiles and received training on finishing the task with a different topic. They were asked to make corrections only after they finished composing a certain length of text. The data corpus consists of the recognition output of their dictations excluding corrections. Word recognition errors were first marked by the participants themselves and then validated by researchers via cross-referencing the recorded audios. The data corpus contains 4,804 words.

5.2 Evaluation Metrics

To evaluate the overall performance of the error detection, classification error rate (CER) (Equation 1), commonly used metric to evaluate classifiers, is used. CER is the percentage of words that are wrongly classified.

$$CER = \frac{\#\ of\ wrongly\ classified\ words}{total\#\ of\ words} \quad (1)$$

The baseline CER is derived by assuming all the words are correct, and it has the value as the ratio of the total number of insertion and substitution errors to the total number of output words.

Precision (PRE) and recall (REC) on errors are used to measure the performance of identifying er-

[2] Available via http://www.ling.gu.se/˜lager/mutbl.html

rors. PRE is the percentage of words classified as errors that are in fact recognition errors. REC denotes the proportion of actual recognition errors that are categorized as errors by the classifier. In addition, F measure (Equation 2), a single-valued metric reflecting the trade-off between PRE and REC, is also used. The baselines of PRE, REC, and F for error are zeros, for all of the output words are assumed correct.

$$F = \frac{2 * PRE * REC}{PRE + REC} \tag{2}$$

5.3 Results

3-fold cross-validation was used to test the system. When dividing the data corpus, sentence is treated as an atomic unit. The 3-fold cross-validation was run 9 times, and the average performance is reported in Table 2. The labels of rule combinations are defined by the connections of several symbols defined in Section 4.2. For each rule combination, the types of rules can be included are decided by all the possible combinations of those symbols which are in Table 1. For example, L-CS-Local-Long includes rules with conditions L, CS, Local, Long, LCS, CSLocal and CSLong.

The threshold of net positive effect is set to 5 to ensure that enough evidence has been observed, and that of the ratio of the positive effect is set to 0.5 to ensure that selected transformations have the positive effects.

For the combinations without CS, L-Local-Long achieves the best performance in terms of both CER and F measure. A relative improvement of 4.85% is achieved over the baseline CER, which is relatively small. One possible explanation concerns the large vocabulary size in the data set. Although the participants were asked to compose the documents on the same topic, the word usage was greatly diversified. An analysis of the data corpus shows that the vocabulary size is 993.

Despite its best performance in linguistic feature groups, L-Local-Long produces worse performance than CS in both CER and F measure. Therefore, linguistic features by themselves are not as useful as confidence scores.

When linguistic features are combined with CS, they provide additional improvement. L-CS achieves a 4.58% relative improvement on CER and a 31.37% relative improvement on F measure over CS. L-CS-Local only achieves marginal improvement on CER and a 7.54% relative improvement on F measure over L-CS.

The best performance is generated by L-CS-Local-Long. In particular, it boosts CER by a relative improvement of 12.30% over CS and a relative improvement of 7.02% over L-CS-Local. In addition, it improves F measure by 53.62% and 8.74% in comparison with CS and L-CS-Local respectively. Therefore, enlarging the scope of context can lead to improved performance on error detection.

It is revealed from Table 2 that the improvement on F measure is due to the improvement on recall without hurting the precision. After combining linguistic features with CS, L-CS and L-CS-Local-Long achieve 43.77% and 75.57% relative improvements on recall over CS separately. Hence, the linguistic features can improve the system's ability in finding more errors. Additionally, L-CS-Local-Long achieves a 7.32% relative improvement on precision over CS.

The average numbers of learned rules are shown in Table 2. With the increased number of possible used pre-defined rules, the number of learned rules increases moderately. L-CS-Local-Long and L-CS-Local have the largest number of rules, 14, which is rather a small set of rules. As discussed above, these rules are straightforward and easy to understand.

Figure 2 shows CERs when the learned rules are incrementally applied in one run for L-CS-Local-Long. Three lines represent each of the three folds separately, and the number of learned rules differs among folds.

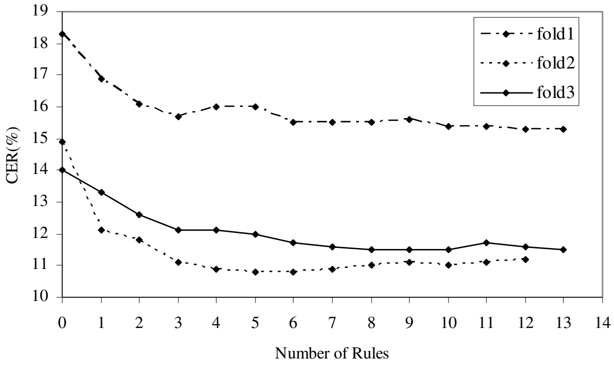

Figure 2: Relations of CERs with Number of Rules

Combination	Mean CER (%)	Std. Dev	Mean PRE (%)	Mean REC (%)	Mean F (%)	Mean # of rules
Baseline	15.66	0.06	-	-	-	-
L	15.55	0.11	61.85	2.04	3.88	3
L-Local	15.58	0.14	60.88	2.19	4.17	4
L-Local-Long	14.90	0.10	61.67	13.83	22.37	8
CS	14.64	0.09	61.03	21.98	31.50	1
L-CS	13.97	0.15	61.48	31.60	41.38	8
L-CS-Local	13.81	0.18	61.28	35.52	44.50	14
L-CS-Local-Long	12.84	0.21	65.50	38.59	48.39	14

Table 2: Performance of Transformation Rule Combinations

After the first several rules are applied, CERs drop significantly. Then the changes in CERs become marginal as additional rules are applied. The fold 1 and 3 reach the lowest CER after the last rule is applied, and fold 2 reaches the lowest CERs in the middle. Thus, the top ranked rules are mostly useful.

One advantage of TBL is that the learning result can be easily interpreted. The following is the top six rules learned in fold 3 in Figure 2.

Mark a word as an error, if :

- its confidence score is less than 0; it is in the middle of a sentence; and it is a null-link word.
- its confidence score is less than -5; it is in the middle of a sentence; and it has links to preceding words.
- its confidence score is less than 0; it is the first word of a sentence; and it is a null-link word.
- its confidence score is less than 2; it is in the middle of a sentence; it has 1 syllable; and the word following it also has 1 syllable and is an error.
- its confidence score is less than -1; and both its preceding and following words are errors.

Mark a word as a correct word, if :

- its confidence score is greater than -1; and both its preceding and following words are correct words.

All of the above six rules include word confidence score as a feature. Rule 1 and rule 3 suggest that null-link words are good indicators of errors, which confirms our hypothesis. Rule 2 shows that a word with low confidence score may also be an error even if it is part of the linkage of the sentence. Rule 4 shows continuous short words are possible errors. Rule 5 indicates that a word with low confidence score may be an error if its surrounding words are errors. Rule 6 is a rule to compensate for the wrongly labeled words by previous rules.

6 Conclusion and Future Works

We introduced an error detection method based on feature combinations. Transformation-based learning was used as the classifier to combine linguistic features with word confidence scores. Two kinds of linguistic features were selected: lexical features extracted from words themselves, and syntactic features from the parsing result of link grammar. Transformation templates were defined by varying scope of the context. Experimental results on a dictation corpus showed that although linguistic features alone were not as useful as word confidence scores to error detection, they provided complementary information when combined with word confidence score. Moreover, the performance of error detection was improved incrementally as the scope of context was enlarged, and the best performance was achieved when sentence context was considered. In particular, enlarging the context modeled by linguistic features improved the capability of error detection by finding more errors without deteriorating and even improving the precision.

The proposed method has been tested using a dictation corpus on a topic related to office environment. We are working on evaluating the method on spontaneous dictation utterances from the CSR-II corpus, and other monologue corpora such as Broadcast News. The method can be extended by incorporating lexical semantic features from the semantic analysis of recognition output to detect semantic errors that are likely overlooked by syntactic analysis.

Acknowledgement

This work is supported by the National Science Foundation under Grant# 0328391. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF).

References

Eric Brill, Radu Florian, John C. Henderson, and Lidia Mangu. 1998. Beyond n-grams: Can linguistic sophistication improve language modeling? In *Proceedings of COLING/ACL*, pages 186–190.

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21(4):543–565.

Paul Carpenter, Chun Jin, Daniel Wilson, Rong Zhang, Dan Bohus, and Alex Rudnicky. 2001. Is this conversation on track? In *Proceedings of Eurospeech*, pages 2121–2124.

Lin L. Chase. 1997. *Error-Responsive Feedback Mechanisms for Speech Recognizers*. Ph.D. thesis, School of Computer Science, CMU, April.

Jinjuan Feng, Andrew Sears, and Clare-Marie Karat. 2003. A longitudinal investigation of hands-free speech based navigation during dictation. Technical report, UMBC.

Dennis Grinberg, John Lafferty, and Daniel Sleator. 1995. A robust parsing algorithm for link grammars. Technical Report CMU-CS-95-125, Carnegie Mellon University.

Torbjörn Lager. 1999. The μ-tbl system: Logic programming tools for transformation-based learning. In *Proceedings of the third international workshop on computational natural language learning*.

Lidia Mangu and Mukund Padmanabhan. 2001. Error corrective mechanisms for speech recognition. In *Proceedings of ICASSP*, volume 1, pages 29–32.

Christine Pao, Philipp Schmid, and James Glass. 1998. Confidence scoring for speech understanding systems. In *Proceedings of ICSLP*, pages 815–818.

Manny Rayner, David Carter, Vassilios Digalakis, and Patti Price. 1994. Combining knowledge sources to reorder n-best speech hypothesis lists. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 212–217.

Rubén San-Segundo, Bryan Pellom, Kadri Hacioglu, and Wayne Ward. 2001. Confidence measures for spoken dialogue systems. In *Proceedings of ICASSP*, volume 1, pages 393–396.

Ruhi Sarikaya, Yuqing Gao, and Michael Picheny. 2003. Word level confidence measurement using semantic features. In *Proceedings of ICASSP*, volume 1, pages 604–607.

Andrew Sears, Clare-Marie Karat, Kwesi Oseitutu, Azfar S. Karimullah, and Jinjuan Feng. 2001. Productivity, satisfaction, and interaction strategies of individuals with spinal cord injuries and traditional users interacting with speech recognition software. *Universal Access in the Information Society*, 1(1):4–15, June.

Gabriel Skantze and Jens Edlund. 2004. Early error detection on word level. In *Proceedings of Robust*.

Daniel Sleator and Davy Temperley. 1993. Parsing english with a link grammar. In *Proceedings of the third international workshop on parsing technologies*.

Rong Zhang and Alexander I. Rudnicky. 2001. Word level confidence annotation using combinations of features. In *Proceedings of Eurospeech*, pages 2105–2108.

Lina Zhou, Yongmei Shi, Jinjuan Feng, and Andrew Sears. 2005. Data mining for detecting errors in dictation speech recognition. *IEEE Transactions on Speech and Audio Processing, Special Issues on Data Mining of Speech, Audio and Dialog*, 13(5), September.