Improving Information Extraction by Modeling Errors in Speech Recognizer Output

David D. Palmer †‡
† The MITRE Corporation
202 Burlington Road
Bedford, MA 01730
palmer@mitre.org

Mari Ostendorf ‡
‡Electrical Engineering Dept.
University of Washington
Seattle, WA 98195
mo@ee.washington.edu

ABSTRACT

In this paper we describe a technique for improving the performance of an information extraction system for speech data by explicitly modeling the errors in the recognizer output. The approach combines a statistical model of named entity states with a lattice representation of hypothesized words and errors annotated with recognition confidence scores. Additional refinements include the use of multiple error types, improved confidence estimation, and multi-pass processing. In combination, these techniques improve named entity recognition performance over a text-based baseline by 28%.

Keywords

ASR error modeling, information extraction, word confidence

1. INTRODUCTION

There has been a great deal of research on applying natural language processing (NLP) techniques to text-based sources of written language data, such as newspaper and newswire data. Most NLP approaches to spoken language data, such as broadcast news and telephone conversations, have consisted of applying text-based systems to the output of an automatic speech recognition (ASR) system; research on improving these approaches has focused on either improving the ASR accuracy or improving the text-based system (or both). However, applying text-based systems to ASR output ignores the fact that there are fundamental differences between written texts and ASR transcriptions of spoken language: the style is different between written and spoken language, the transcription conventions are different, and, most importantly, there are errors in ASR transcriptions. In this work, we focus on the third problem: handling errors by explicitly modeling uncertainty in ASR transcriptions.

.

The idea of explicit error handling in information extraction (IE) from spoken documents was introduced by Grishman in [1], where a channel model of word insertions and deletions was used with a deterministic pattern matching system for information extraction. While the use of an error model resulted in substantial performance improvements, the overall performance was still quite low, perhaps because the original system was designed to take advantage of orthographic features. In looking ahead, Grishman suggests that a probabilistic approach might be more successful at handling errors.

The work described here provides such an approach, but introduces an acoustically-driven word confidence score rather than the word-based channel model proposed in [1]. More specifically, we provide a unified approach to predicting and using uncertainty in processing spoken language data, focusing on the specific IE task of identifying named entities (NEs). We show that by explicitly modeling multiple types of errors in the ASR output, we can improve the performance of an IE system, which benefits further from improved error prediction using new features derived from multi-pass processing.

The rest of the paper is organized as follows. In Section 2 we describe our error modeling, including explicit modeling of multiple ASR error types. New features for word confidence estimation and the resulting performance improvement is given in Section 3. Experimental results for NE recognition are presented in Section 4 using Broadcast News speech data. Finally, in Section 5, we summarize the key findings and implications for future work.

2. APPROACH

Our approach to error handling in information extraction involves using probabilistic models for both information extraction and the ASR error process. The component models and an integrated search strategy are described in this section.

2.1 Statistical IE

We use a probabilistic IE system that relates a word sequence $W = w_1, \ldots, w_M$ to a sequence of information states $S = s_1, \ldots, s_M$ that provide a simple parse of the word sequence into phrases, such as name phrases. For the work described here, the states s_t correspond to different types of NEs. The IE model is essentially a phrase language model:

$$p(S,W) = p(s_1,\ldots,s_M,w_1,\ldots,w_M) \quad (1)$$
$$= \prod_{t=1}^{M} p(w_t|w_{t-1},s_t)p(s_t|s_{t-1},w_{t-1})$$

with state-dependent bigrams $p(w_t|w_{t-1},s_t)$ that model the types of words associated with a specific type of NE, and state transition probabilities $p(s_t|s_{t-1},w_{t-1})$ that mix the Markov-like structure of an HMM with dependence on the previous word. (Note that titles, such as "President" and "Mr.", are good indicators of transition to a name state.)

This IE model, described further in [2], is similar to other statistical approaches [3, 4] in the use of state dependent bigrams, but uses a different smoothing mechanism and state topology. In addition, a key difference in our work is explicit error modeling in the "word" sequence, as described next.

2.2 Error Modeling

To explicitly model errors in the IE system, we introduce new notation for the hypothesized word sequence, $H = h_1,\ldots,h_M$, which may differ from the actual word sequence W, and a sequence of error indicator variables $K = k_1,\ldots,k_M$, where $k_t = 1$ when h_t is an error and $k_t = 0$ when h_t is correct. We assume that the hypothesized words from the recognizer are each annotated with confidence scores

$$\gamma_t = p(k_t = 0|H,A) = p(h_t = w_t|H,A),$$

where A represents the set of features available for initial confidence estimation from the recognizer, acoustic or otherwise.

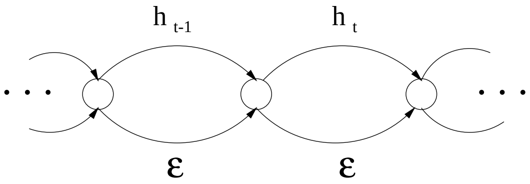

Figure 1: Lattice with correct and error paths.

We construct a simple lattice from $h_1,\ldots,h_M$ with "error" arcs indicated by ϵ-tokens in parallel with each hypothesized word h_t, as illustrated in Figure 1. We then find the maximum posterior probability state sequence by summing over all paths through the lattice:

$$S^* = \operatorname*{argmax}_S p(S|H,A), \quad (2)$$
$$= \operatorname*{argmax}_S \sum_K p(S,K|H,A) \quad (3)$$

or, equivalently, marginalizing over the sequence K. Equation 3 thus defines the decoding of named entities via the state sequence S, which (again) provides a parse of the word sequence into phrases.

Assuming first that K and H encode all the information from A about S, and then that the specific value h_t occurring at an error does not provide additional information for the NE states[1] S, we can rewrite Equation 3 as:

$$S^* = \operatorname*{argmax}_S \sum_K p(K|H,A)p(S|K,H,A)$$
$$= \operatorname*{argmax}_S \sum_K p(K|H,A)p(S|K,H)$$
$$= \operatorname*{argmax}_S \sum_K p(K|H,A)p(S|W_{(K,H)}).$$

For the error model, $p(K|H,A)$, we assume that errors are conditionally independent given the hypothesized word sequence H and the evidence A:

$$p(K|H,A) = \prod_{t=1}^{M} p(k_t|H,A). \quad (4)$$

where $\gamma_t = p(k_t = 0|H,A)$ is the ASR word "confidence". Of course, the errors are not independent, which we take advantage of in our post-processing of confidence estimates, described in Section 3.

We can find $p(S|W)$ directly from the information extraction model, $p(S,W)$ described in Section 2.1, but there is no efficient decoding algorithm. Hence we approximate

$$p(S|W) = \frac{p(S,W)}{p(W)} \approx \bar{p}(S,W) \quad (5)$$

assuming that the different words that could lead to an error are roughly uniform over the likely set. More specifically, $\bar{p}(S,W)$ incorporates a scaling term as follows:

$$\bar{p}(\epsilon|w_{t-1} = v, s_t) = \frac{1}{N_v} p(\epsilon|w_{t-1} = v, s_t) \quad (6)$$

where N_v is the number of different error words observed after v in the training set and $p(\epsilon|v,s_t)$ is trained by collapsing all different errors into a single label ϵ. Training this language model requires data that contains ϵ-tokens, which can be obtained by aligning the reference data and the ASR output. In fact, we train the language model with a combination of the original reference data and a duplicate version with ϵ-tokens replacing error words.

Because of the conditional independence assumptions behind equations 1 and 4, there is an efficient algorithm for solving equation 3, which combines steps similar to the forward and Viterbi algorithms used with HMMs. The search is linear with the length M of the hypothesized word sequence and the size of the state space (the product space of NE states and error states). The forward component is over the error state (parallel branches in the lattice), and the Viterbi component is over the NE states.

If the goal is to find the words that are in error (e.g. for subsequent correction) as well as the named entities, then the objective is

$$(S,K)^* = \operatorname*{argmax}_{S,K} p(S,K|H,A) \quad (7)$$
$$\approx \operatorname*{argmax}_{S,K} p(K|H,A)\bar{p}(S,W_{(K,H)}), \quad (8)$$

[1] Clearly, some hypotheses do provide information about S in that a reasonably large number of errors involve simple ending differences. However, our current system has no mechanism for taking advantage of this information explicitly, which would likely add substantially to the complexity of the model.

which simply involves finding the best path K^* through the lattice in Figure 1. Again because of the conditional independence assumption, an efficient solution involves Viterbi decoding over an expanded state space (the product of the names and errors). The sequence K^* can help us define a new word sequence $\hat{W}$ that contains ϵ-tokens: $\hat{w}_t = h_t$ if ${k_t}^* = 0$, and $\hat{w}_t = \epsilon$ if ${k_t}^* = 1$. Joint error and named entity decoding results in a small degradation in named entity recognition performance, since only a single error path is used. Since errors are not used explicitly in this work, all results are based on the objective given by equation 3.

Note that, unlike work that uses confidence scores γ_t as a weight for the hypothesized word in information retrieval [5], here the confidence scores also provide weights $(1 - \gamma_t)$ for explicit (but unspecified) sets of alternative hypotheses.

2.3 Multiple Error Types

Though the model described above uses a single error token ϵ and a 2-category word confidence score (correct word vs. error), it is easily extensible to multiple classes of errors simply by expanding the error state space. More specifically, we add multiple parallel arcs in the lattice in Figure 1, labeled ϵ_1, ϵ_2, etc., and modify confidence estimation to predict multiple categories of errors.

In this work, we focus particularly on distinguishing out-of-vocabulary (OOV) errors from in-vocabulary (IV) errors, due to the large percentage of OOV words that are names (57% of OOVs occur in named entities). Looking at the data another way, the percentage of name words that are OOV is an order of magnitude larger than words in the “other” phrase category, as described in more detail in [6]. As it turns out, since OOVs are so infrequent, it is difficult to robustly estimate the probability of IV vs. OOV errors from standard acoustic features, and we simply use the relative prior probabilities to scale the single error probability.

3. CONFIDENCE PREDICTION

An essential component of our error model is the word-level confidence score, $p(k_t|H, A)$, so one would expect that better confidence scores would result in better error modeling performance. Hence, we investigated methods for improving the confidence estimates, focusing specifically on introducing new features that might complement the features used to provide the baseline confidence estimates. The baseline confidence scores used in this study were provided by Dragon Systems. As described in [7], the Dragon confidence predictor used a generalized linear model with six inputs: the word duration, the language model score, the fraction of times the word appears in the top 100 hypotheses, the average number of active HMM states in decoding for the word, a normalized acoustic score and the log of the number of recognized words in the utterance. We investigated several new features, of which the most useful are listed below.

First, we use a short window of the original confidence scores: γ_t, γ_{t-1} and γ_{t+1}. Note that the post-processing paradigm allows us to use non-causal features such as γ_{t+1}. We also define three features based on the ratios of γ_{t-1}, γ_t, and γ_{t+1} to the average confidence for the document in which h_t appears, under the assumption that a low confidence score for a word is less likely to indicate a word error if the average confidence for the entire document is also low. We hypothesized that words occurring frequently in a large window would be more likely to be correct, again assuming that the ASR system would make errors randomly from a set of possibilities. Therefore, we define features based on how many times the hypothesis word h_t occurs in a window $(h_{t-n}, ..., h_t, ..., h_{t+n})$ for $n = 5$, 10, 25, 50, and 100 words. Finally, we also use the relative frequency of words occurring as an error in the training corpus, again looking at a window of ± 1 around the current word.

Due to the close correlation between names and errors, we would expect to see improvement in the error modeling performance by including information about which words are names, as determined by the NE system. Therefore, in addition to the above set of features, we define a new feature: *whether the hypothesis word h_t is part of a location, organization, or person phrase*. We can determine the value of this feature directly from the output of the NE system. Given this additional feature, we can define a multi-pass processing cycle consisting of two steps: confidence re-estimation and information extraction. To obtain the name information for the first pass, the confidence scores are re-estimated without using the name features, and these confidences are used in a joint NE and error decoding system. The resulting name information is then used, in addition to all the features used in the previous pass, to improve the word confidence estimates. The improved confidences are in turn used to further improve the performance of the NE system.

We investigated three different methods for using the above features in confidence estimation: decision trees, generalized linear models, and linear interpolation of the outputs of the decision tree and generalized linear model. The decision trees and generalized linear models gave similar performance, and a small gain was obtained by interpolating these predictions. For simplicity, the results here use only the decision tree model.

A standard method for evaluating confidence prediction [8] is the normalized cross entropy (NCE) of the binary correct/error predictors, that is, the reduction in uncertainty in confidence prediction relative to the ASR system error rate. Using the new features in a decision tree predictor, the NCE score of the binary confidence predictor improved from 0.195 to 0.287. As shown in the next section, this had a significant impact on NE performance. (See [6] for further details on these experiments and an analysis of the relative importance of different factors.)

4. EXPERIMENTAL RESULTS

The specific information extraction task we address in this work is the identification of name phrases (names of persons, locations, and organizations), as well as identification of temporal and numeric expressions, in the ASR output. Also known as named entities (NEs), these phrases are useful in many language understanding tasks, such as coreference resolution, sentence chunking and parsing, and summarization/gisting.

4.1 Data and Evaluation Method

The data we used for the experiments described in this paper consisted of 114 news broadcasts automatically annotated with recognition confidence scores and hand labeled with NE types and locations. The data represents an intersection of the data provided by Dragon Systems for the 1998 DARPA-sponsored Hub-4 Topic, Detection and Tracking (TDT) evaluation and those stories for which named entity labels were available. Broadcast news data is particularly appropriate for our work since it contains a high density of name phrases, has a relatively high word error rate, and requires a virtually unlimited vocabulary.

We used two versions of each news broadcast: a reference transcription prepared by a human annotator and an ASR transcript prepared by Dragon Systems for the TDT evaluation [7]. The Dragon ASR system had a vocabulary size of about 57,000 words and a word error rate (WER) of about 30%. The ASR data contained the word-level confidence information, as described earlier, and the reference transcription was manually-annotated with named entity information. By aligning the reference and ASR transcriptions, we were able to determine which ASR output words corresponded to errors and to the NE phrases.

We randomly selected 98 of the 114 broadcasts as training data, 8 broadcasts as development test, and 8 broadcasts as evaluation test data, which were kept "blind" to ensure unbiased evaluation results. We used the training data to estimate all model parameters, the development test set to tune parameters during development, and the evaluation test set for all results reported here. For all experiments we used the same training and test data.

4.2 Information Extraction Results

Table 1 shows the performance of the baseline information extraction system (row 1) which does not model errors, compared to systems using one and two error types, with the baseline confidence estimates and the improved confidence estimates from the previous section. Performance figures are the standard measures used for this task: F-measure (harmonic mean of recall and precision) and slot error rate (SER), where separate type, extent and content error measures are averaged to get the reported result.

The results show that modeling errors gives a significant improvement in performance. In addition, there is a small but consistent gain from modeling OOV vs. IV errors separately. Further gain is provided by each improvement to the confidence estimator.

Since the evaluation criterion involves a weighted average of content, type and extent errors, there is an upper bound of 86.4 for the F-measure given the errors in the recognizer output. In other words, this is the best performance we can hope for without running additional processing to correct the ASR errors. Thus, the combined error modeling improvements lead to recovery of 28% of the possible performance gains from this scheme. It is also interesting to note that the improvement in identifying the extent of a named entity actually results in a decrease in performance of the content component, since words that are incorrectly recognized are introduced into the named entity regions.

5. DISCUSSION

In this paper we described our use of error modeling to improve information extraction from speech data. Our model is the first to explicitly represent the uncertainty inherent in the ASR output word sequence. Two key innovations are the use of word confidence scores to characterize the ASR outputs and alternative hypotheses, and integration of the error model with a statistical model of information extraction. In addition, improvements in performance were obtained by modeling multiple types of errors (in vocabulary vs. out of vocabulary) and adding new features to the confidence estimator obtained using multipass processing. The new features led to improved confidence estimation from a baseline NCE of 0.195 to a value of 0.287. The use of the error model with these improvements resulted in a reduction in slot error rate of 14% and an improvement in the F-measure from 68.4 to 73.4.

Table 1: *Named entity (NE) recognition results using different error models and feature sets for predicting confidence scores. The baseline confidence scores are from the Dragon recognizer, the secondary processing re-estimates confidences as a function of a window of these scores, and the names are provided by a previous pass of named entity detection.*

ϵ-tokens	Confidence Scores	NE F-Measure	NE SER
none	none	68.4	50.9
1	baseline	71.4	46.1
2	baseline	71.5	45.9
1	+ secondary	71.8	44.9
2	+ secondary	72.0	44.8
1	+ secondary + names	73.1	44.3
2	+ secondary + names	73.4	43.9

The integrated model can be used for recognition of NE's alone, as in this work, or in joint decoding of NEs and errors. Since ASR errors substantially degrade NE recognition rates (perfect NE labeling with the errorful outputs here would have an F-measure of 86.4), and since many names are recognized in error because they are out of the recognizer's vocabulary, an important next step in this research is explicit error detection and correction. Preliminary work in this direction is described in [6]. In addition, while this work is based on 1-best recognition outputs, it is straightforward to use the same algorithm for lattice decoding, which may also provide improved NE recognition performance.

Acknowledgments

The authors thank Steven Wegmann of Dragon Systems for making their ASR data available for these experiments and BBN for preparing and releasing additional NE training data. This material is based in part upon work supported by the National Science Foundation under Grant No. IIS0095940. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

6. REFERENCES

[1] R. Grishman, "Information extraction and speech recognition," *Proceedings of the Broadcast News*

Transcription and Understanding Workshop, pp. 159–165, 1998.

[2] D. Palmer, M. Ostendorf, and J. Burger ‘Robust Information Extraction from Automatically Generated Speech Transcriptions,” *Speech Communication,* vol. 32, pp. 95–109, 2000.

[3] D. Bikel, R. Schwartz, R. Weischedel, “An Algorithm that Learns What’s in a Name,” *Machine Learning,* 34(1/3):211–231, 1999.

[4] Y. Gotoh, S. Renals, “Information Extraction From Broadcast News,”*Philosophical Transactions of the Royal Society,* series A: Mathematical, Physical and Engineering Sciences, 358(1769):1295–1308, 2000.

[5] A. Hauptmann, R. Jones, K. Seymore, S. Slattery, M. Witbrock, and M. Siegler, “Experiments in information retrieval from spoken documents,” *Proceedings of the Broadcast News Transcription and Understanding Workshop,* pp. 175–181, 1998.

[6] D. Palmer, *Modeling Uncertainty for Information Extraction from Speech Data,* Ph.D. dissertation, University of Washington, 2001.

[7] L. Gillick, Y. Ito, L. Manganaro, M. Newman, F. Scattone, S. Wegmann, J. Yamron, and P. Zhan, “Dragon Systems’ Automatic Transcription of New TDT Corpus,” *Proceedings of the Broadcast News Transcription and Understanding Workshop,* pp. 219–221, 1998.

[8] M. Siu and H. Gish, “Evaluation of word confidence for speech recognition systems,” *Computer Speech & Language,* 13(4):299–319, 1999.