Discriminative Named Entity Recognition of Speech Data using Speech Recognition Confidence

Katsuhito Sudoh, Hajime Tsukada, and Hideki Isozaki

NTT Communication Science Labs.
2-4 Hikaridai, Keihanna Science City, Kyoto 619-0237, Japan
{sudoh,tsukada,isozaki}@cslab.kecl.ntt.co.jp

Abstract

This paper presents a method for the named entity recognition (NER) of speech data that uses automatic speech recognition (ASR) confidence as a feature that indicates whether each word is correctly recognized. An NER model is trained using ASR results with named entity (NE) labels to include an ASR confidence feature as well as corresponding transcriptions with NE labels. Experiments using support vector machines (SVMs) and speech data from Japanese newspaper articles show that the proposed method achieves higher F-measure in NER than a simple application of text-based NER to ASR results.

Index Terms: named entity recognition, speech recognition, confidence scoring, discriminative models, information retrieval.

1. Introduction

These days we can obtain a large volume of information from all over the world because of the growth of network bandwidths and storage capacities. Text data such as newspaper articles and WWW pages are major information sources used for natural language processing (NLP) applications including information extraction, question answering, and summarization. On the other hand, speech data such as broadcast news and PodCasts are also becoming important information sources. We aim to use speech data for such NLP applications as DARPA's global autonomous language exploitation (GALE) program.

Named entities (NEs) are expressions that usually consist of compound words, such as peoples' names, location names, and temporal entities (date and time). Since NEs hold important information, named entity recognition (NER) is one of the key techniques for NLP applications. In this paper, we focus on the NER of automatic speech recognition (ASR) results, in which we face ASR errors due to out-of-vocabulary (OOV) words and mismatch between acoustic/language models and inputs, even with state-of-the-art technologies. Although continuous efforts to improve ASR accuracy are needed, developing a robust NER for noisy word sequences containing ASR errors is also important.

Most previous studies on the NER of speech data used generative models such as hidden Markov models (HMMs) [1–5]. A problem with generative models is that non-independent features are difficult to use [6]. In NER, various features are effective for determining whether a word is a part of an NE: whether the word is a noun, whether the word starts with a capital letter, and whether the word is at the beginning of a sentence, however, such features are not independent of each other. For this reason, recent studies on text-based NER use discriminative models such as maximum entropy (ME) models [7, 8], support vector machines (SVMs) [9], and conditional random fields (CRFs) [10] with those non-independent features. Zhai *et al.* [11] applied such a text-based NER method to ASR results. A problem of applying text-based NER is that ASR errors cause erroneous words to be extracted as NEs, decreasing NER precision. To address the problem, Palmer and Ostendorf [2] modeled ASR errors using ASR confidence to reject erroneous ASR word hypotheses in the NER of speech data based on a generative model. This rejection helps avoid the extraction of erroneous NEs and is especially effective when ASR accuracy is relatively low. However, such effective but non-independent features are hard to use in their generative model.

We extend their approach to discriminative models and propose an NER method that deals with ASR confidence as a feature. The method enables ASR error rejection in NER and can also use various effective features that may not be independent of each other. In experiments using SVM-based NER and speech data from Japanese newspaper articles, the ASR confidence feature increased the NER F-measure, especially in precision, compared to simply applying text-based NER to the ASR results.

2. SVM-based NER

NER is a task that identifies NEs in documents and labels their name categories. For example, in the phrase "The prime minister of Japan, Jun-ichiro Koizumi ...," the word *Japan* is labeled an NE of a location (country) and the compound word *Jun-ichiro Koizumi* is labeled an NE of a person. We solve this chunking problem by classifying words into NE classes that consist of NE categories (such as `PERSON`, `LOCATION`) and chunking states. We used four chunking states for each NE based on a Start/End method [12]: `BEGIN` (beginning of an NE), `MIDDLE` (middle of an NE), `END` (ending of an NE), and `SINGLE` (a single word NE). For example, the class of a word at the beginning of a person's name is denoted as `PERSON-BEGIN`. If a word does not constitute an NE, it is classified into the non-NE class `OTHER`.

In this paper, we employ an SVM-based NER method [9] that showed good NER performance in Japanese. We use three features for each word: the word itself, part-of-speech tag, and character type. For context dependence, we also use those features for the two preceding and succeeding words. Each feature is represented by a binary value (1 or 0), for example, "whether the previous word is *Japan*," "whether the part-of-speech of the current word is *particle*," and "whether the character type of the next word is `all-capital`." Then, each word is classified based on a long binary vector, where only 15 elements are 1 and the others are 0. Using SVMs for NER has two problems. One, SVMs can only solve two-class problems. We reduce the multi-

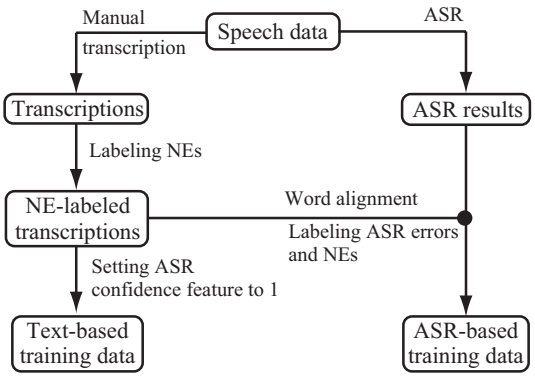

Figure 1: Procedure for preparing training data

Table 1: Text-based training data

Word	Confidence	NE label
Nomo	1	PERSON-BEGIN
Hideo	1	PERSON-END
tohshu	1	OTHER
no	1	OTHER
America	1	LOCATION-SINGLE

Table 2: ASR-based training data

Word	Confidence	NE label
Noro	0	OTHER
Hideo	1	OTHER
tohshu	1	OTHER
ni	0	OTHER
America	1	LOCATION-SINGLE

class problems of NER to a group of two-class problems using a *one-against-all* approach where each SVM is trained to distinguish members of a class (e.g., LOCATION-BEGIN) from non-members (PERSON-BEGIN, LOCATION-MIDDLE, ...). In this approach, two or more classes may be assigned to a word or no class may be assigned to a word. To avoid these situations, we choose class c that has the largest SVM output score $g_c(x)$ among all others. The other problem is that chunking states of the NE labels may not be consistent after classifying all words in a sentence; for example, ARTIFACT-END may follow LOCATION-BEGIN. To maintain the consistency of chunking states, we use a Viterbi search to obtain the best and consistent NE label sequence based on the probability-like values obtained by applying sigmoid function $s_n(x) = 1/(1 + \exp(-x))$ to SVM output scores.

3. Proposed method

3.1. Incorporating ASR confidence into NER

A problem in the NER of ASR results is that ASR errors cause erroneous NEs. We cannot recognize an NE when there are ASR errors in one or more words constituting the NE. Even if none of the words constituting the NE have ASR errors, we may not be able to recognize the NE correctly due to ASR errors in context words. To avoid these ASR error problems, in this paper we model ASR errors in NER using an additional feature, which indicates whether each word is correctly recognized, called the *ASR confidence feature*. Our NER model is trained using ASR results with the ASR confidence feature, and we estimate feature values using ASR confidence scores in testing.

Note that we only aim to identify NEs whose words are correctly recognized by ASR. NEs containing ASR errors are ignored in our method, because identifying such NEs, especially those containing out-of-vocabulary (OOV) words, is an important but a more difficult problem beyond the scope of this paper.

3.2. Training NER model

Figure 1 illustrates the procedure for preparing the training data for our NER model from speech data. First, speech data are manually transcribed and automatically recognized by ASR. Second, NEs in the transcriptions are labeled, and then the ASR confidence feature values are set to 1. These feature values mean that all the words in the transcriptions are regarded as correctly recognized words. Finally, the ASR results are aligned to the transcriptions to identify ASR errors and NEs. The ASR confidence feature values of correctly recognized words are set to 1 and misrecognized words are set to 0. Each NE is labeled if and only if all words constituting the NE are correctly recognized; otherwise the NE is ignored and those words are labeled OTHER. Tables 1 and 2 show examples of text-based and ASR-based training data. Since the name *Nomo Hideo* in Table 1 is misrecognized in ASR, the correctly recognized word *Hideo* is also labeled OTHER in Table 2.

3.3. ASR confidence scoring

ASR confidence scoring is an important technique in many ASR applications. There are two major approaches for ASR confidence scoring: using a single confidence measure such as word posterior probabilities on word graphs [13], and integrating several confidence measures using classifiers such as neural networks [14], linear discriminant analysis [15], and SVMs [16].

We use SVMs for ASR confidence scoring in this paper to achieve a better performance than with only word posterior probabilities. SVMs are trained using ASR results whose errors are known through their alignment to their reference transcriptions, as described in 3.2. The features used for confidence scoring include the word itself, its part-of-speech tag, its word posterior probability, and the two preceding and following words. The word itself and its part-of-speech are represented by a set of binary values, the same as with an SVM-based NER. Since all other features are binary, we reduce real-valued word posterior probability p to ten binary features (if $0 < p \leq 0.1$, if $0.1 < p \leq 0.2$, ... , and if $0.9 < p \leq 1.0$), for simplicity. Although a large variety of features have been proposed in previous studies, we use only these features and save the others for further studies.

SVM output scores are normalized with a sigmoid function $s_w(x) = 1/(1+\exp(-x))$, and the normalized scores are used as ASR confidence scores, which are then used to estimate whether each word is correctly recognized. If the ASR confidence score of a word is greater than threshold t_w, the word is deemed correct and we set the ASR confidence feature value to 1; otherwise we set it to 0.

4. Experiments

To investigate the effect of incorporating ASR confidence into SVM-based NER, we performed the following experiments.

4.1. Setup

We simulated the procedure described in 3.2 using the speech data from a NE-labeled text corpus. We used the training data set of the Information Retrieval and Extraction Exercise (IREX) workshop [17]. It consisted of 1,174 Japanese newspaper articles (10,718 sentences) and about 19,000 NEs in eight categories (artifact, organization, location, person, date, time, money, and percent). The sentences were read by 106 speakers (about 100 sentences per speaker), and the recorded speech data were used for the experiments. The experiments were conducted with 5-fold cross validation, using 80% of the 1,174 articles and the ASR results of the speech data for training SVMs (both for ASR confidence scoring and for NER) and the rest for the test.

We tokenized the sentences into words and tagged the part-of-speech information using the Japanese morphological analyzer ChaSen 2.3.3 [18] and then labeled the NEs. After tokenization and removing unreadable tokens such as parentheses, the text corpus had 264,388 words of 60 part-of-speech types. Since different types of characters are used in Japanese, we used the following character types as features: `single-kanji` (words written with a single Chinese character), `all-kanji` (longer words written in Chinese characters), `hiragana` (words written in *hiragana* Japanese phonograms), `katakana` (words written in *katakana* Japanese phonograms), `number`, `single-capital` (words with a single capitalized letter), `all-capital`, `capitalized` (only the first letter is capitalized), `roman` (other roman character words), and `others` (all other words). We used all the features that appeared in each training set (no feature selection performed). There were 33 NE classes (eight categories * four chunking states + `OTHER`). For NER, we used an SVM-based chunk annotator YamCha 0.33 [19] with a quadratic kernel $(1 + \vec{x} \cdot \vec{y})^2$.

We used a WFST-based ASR engine [20]. The acoustic model was a triphone HMMs, trained using other read speech data of about 50 hours. The language model was a word trigram model with Witten-Bell discounting, trained using other Japanese newspaper articles (about 340 M words) that were also tokenized using ChaSen. The vocabulary size of the language model was 426,023. The number of OOV words in the text corpus was 1,551 (0.587%). 223 (1.23%) NEs in the text corpus contained such OOV words. The word accuracy obtained by our ASR engine for the overall dataset was 79.45%. 82.00% of the NEs remained in the ASR results. Figure 2 shows the ROC curves of ASR error estimation for the five test sets overall, using SVM-based ASR confidence scoring (with the quadratic kernel) and word posterior probabilities as ASR confidence scores, where

$$\text{True positive rate} = \frac{\text{\# correctly recognized words estimated as correct}}{\text{\# correctly recognized words}}$$

$$\text{False positive rate} = \frac{\text{\# misrecognized words estimated as correct}}{\text{\# misrecognized words}}.$$

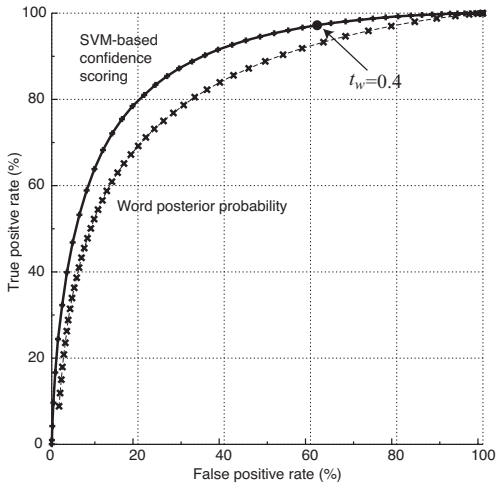

Figure 2: SVM-based confidence scoring outperforms word posterior probability for ASR error estimation.

4.2. Evaluation metrics

Evaluation was based on an averaged NER F-measure, which is the harmonic mean of NER precision and recall:

$$\text{NER precision} = \frac{\text{\# correctly recognized NEs}}{\text{\# recognized NEs}}$$

$$\text{NER recall} = \frac{\text{\# correctly recognized NEs}}{\text{\# NEs in original text}}.$$

The correctness of each recognized NE was identified by the same procedure as used in preparing the ASR-based training data (described in 3.2), considering ASR and name category correctness.

4.3. Compared methods

We compared the following five methods:
Baseline applies text-based NER trained using the original text to the 1-best ASR results.
WordReject rejects unconfident words whose ASR confidence scores are lower than threshold t_w and replaces them with unknown word symbols before applying NER as in Baseline.
Proposed incorporates the ASR confidence feature based on SVM-based ASR confidence scoring.
UpperBound assumes a perfect ASR confidence scoring. The ASR errors in the test set are known, but the NER model is the same as Proposed. This is regarded as the upper-boundary of Proposed.
Reference applies text-based NER trained using the original text to reference transcriptions, assuming word accuracy is 100.0%.

4.4. NER Results

Table 3 summarizes NER results. Proposed showed the best F-measure, 69.02%, which was better than Baseline by 2.0%, from a 7.5% improvement in precision, instead of a recall decrease of 1.9%. WordReject showed worse results than Proposed, with a 1.0% improvement in F-measure over Baseline. Figure 3 shows NER results by WordReject and Proposed by varying word rejection threshold t_w. Proposed outperformed WordReject with any t_w. With $t_w = 0.4$, we obtained the best results in F-measure, which are shown in Table 3.

Table 3: NER results in averaged NER F-measure, precision, and recall. ASR word accuracy was 79.45 %, and 82.00% of NEs remained in ASR results.

Method	F-measure	Precision	Recall
Baseline	67.00%	70.67%	63.70%
WordReject	68.07%	75.93%	61.68%
Proposed	**69.02%**	**78.13%**	**61.81%**
UpperBound	73.14%	87.51%	62.83%
Reference	84.04%	86.27%	81.93%

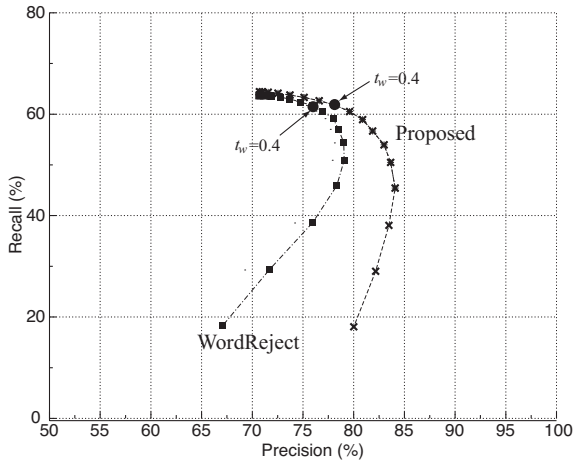

Figure 3: NER precision and recall with varying word rejection threshold t_w.

5. Discussion

The ASR confidence feature effectively improves NER performance mainly in precision, as shown by the difference between Proposed and Baseline in Table 3. The improvement of precision comes from the rejection of NEs based on ASR confidence. Rejection is achieved by identifying *unconfident* words whose ASR confidence scores are lower than threshold t_w, as `OTHER` in NER. The ASR confidence feature is especially expected to work better when ASR accuracy is lower.

Compared to the proposed method, word-level rejection in the application of a text-based NER model to ASR results was less effective, which suggests that modeling ASR errors in the NER model is also effective.

In addition, the difference between UpperBound and Proposed, 4.1% in F-measure, indicated that NER performance can be improved with better ASR confidence scoring.

6. Conclusion

We proposed an NER method for speech data that incorporates ASR confidence as a feature of discriminative NER. The ASR confidence feature is obtained by ASR confidence scoring using several features including word posterior probabilities. In experiments using SVMs, the proposed method shows a higher NER F-measure, especially in terms of improving precision, than simply applying text-based NER to ASR results. The ASR confidence feature rejects erroneous NEs due to ASR errors and is effective for high-precision IE, especially with low ASR accuracy.

For further improvement, we will extend the method to N-best or word lattice inputs [11] and introduce more speech-specific features such as word durations and prosodic features. Future work also includes applying the ASR confidence feature to other tasks in spoken language processing. Since confidence itself is not limited to speech, our approach can also be applied to other noisy inputs, such as optical character recognition (OCR).

7. References

[1] D. Miller, R. Schwartz, R. Weischedel, and R. Stone, "Named entity extraction from broadcast news," in *Proceedings of the DARPA Broadcast News Workshop*, 1999, pp. 37–40.

[2] D. D. Palmer and M. Ostendorf, "Improving information extraction by modeling errors in speech recognizer output," in *Proc. HLT*, 2001.

[3] J. Horlock and S. King, "Named entity extraction from word lattices," in *Proc. EUROSPEECH*, 2003, pp. 1265–1268.

[4] F. Béchet, A. L. Gorin, J. H. Wright, and D. Hakkani-Tür, "Detecting and extracting named entities from spontaneous speech in a mixed-initiative spoken dialogue context: How May I Help You?," *Speech Communication*, vol. 42, no. 2, pp. 207–225, 2004.

[5] B. Favre, F. Béchet, and P. Nocéra, "Robust named entity extraction from large spoken archives," in *Proc. HLT-EMNLP*, 2005, pp. 491–498.

[6] A. McCallum, D. Freitag, and F. Pereira, "Maximum entropy markov models for information extraction and segmentation," in *Proc. ICML*, 2000, pp. 591–598.

[7] A. Borthwick, *A Maximum Entropy Approach to Named Entity Recognition*, Ph.D. thesis, New York University, 1999.

[8] H. L. Chieu and H. T. Ng, "Named entity recognition with a maximum entropy approach," in *Proc. CoNLL*, 2003, pp. 160–163.

[9] H. Isozaki and H. Kazawa, "Efficient support vector classifiers for named entity recognition," in *Proc. COLING*, 2002, pp. 390–396.

[10] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," in *Proc. CoNLL*, 2003.

[11] L. Zhai, P. Fung, R. Schwartz, M. Carpuat, and D. Wu, "Using N-best lists for named entity recognition from chinese speech," in *Proc. HLT-NAACL*, 2004, pp. 37–40.

[12] S. Sekine, R. Grishman, and H. Shinnou, "A decision tree method for finding and classifying names in Japanese texts," in *Proc. the Sixth Workshop on Very Large Corpora*, 1998, pp. 171–178.

[13] F. Wessel, R. Schlüter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 3, pp. 288–298, 2001.

[14] T. Schaaf and T. Kemp, "Confidence measures for spontaneous speech recognition," in *Proc. ICASSP*, 1997, vol. II, pp. 875–878.

[15] S. O. Kamppari and T. J. Hazen, "Word and phone level acoustic confidence scoring," in *Proc. ICASSP*, 2000.

[16] R. Zhang and A. I. Rudnicky, "Word level confidence annotation using combinations of features," in *Proc. EUROSPEECH*, 2001, pp. 2105–2108.

[17] S. Sekine and Y. Eriguchi, "Japanese named entity extraction evaluation - analysis of results," in *Proc. COLING*, 2000, pp. 25–30.

[18] http://chasen.naist.jp/hiki/ChaSen/ (in Japansese).

[19] http://www.chasen.org/˜taku/software/yamcha/.

[20] T. Hori, C. Hori, and Y. Minami, "Fast on-the-fly composition for weighted finite-state transducers in 1.8 million-word vocabulary continuous-speech recognition," in *Proc. ICSLP*, 2004, vol. 1, pp. 289–292.