Incorporating speech recognition confidence into discriminative named entity recognition of speech data

Katsuhito Sudoh Hajime Tsukada Hideki Isozaki
NTT Communication Science Laboratories
Nippon Telegraph and Telephone Corporation
2-4 Hikaridai, Seika-cho, Keihanna Science City, Kyoto 619-0237, Japan
{sudoh,tsukada,isozaki}@cslab.kecl.ntt.co.jp

Abstract

This paper proposes a named entity recognition (NER) method for speech recognition results that uses confidence on automatic speech recognition (ASR) as a feature. The ASR confidence feature indicates whether each word has been correctly recognized. The NER model is trained using ASR results with named entity (NE) labels as well as the corresponding transcriptions with NE labels. In experiments using support vector machines (SVMs) and speech data from Japanese newspaper articles, the proposed method outperformed a simple application of text-based NER to ASR results in NER F-measure by improving precision. These results show that the proposed method is effective in NER for noisy inputs.

1 Introduction

As network bandwidths and storage capacities continue to grow, a large volume of speech data including broadcast news and PodCasts is becoming available. These data are important information sources as well as such text data as newspaper articles and WWW pages. Speech data as information sources are attracting a great deal of interest, such as DARPA's global autonomous language exploitation (GALE) program. We also aim to use them for information extraction (IE), question answering, and indexing.

Named entity recognition (NER) is a key technique for IE and other natural language processing tasks. Named entities (NEs) are the proper expressions for things such as peoples' names, locations' names, and dates, and NER identifies those expressions and their categories. Unlike text data, speech data introduce automatic speech recognition (ASR) error problems to NER. Although improvements to ASR are needed, developing a robust NER for noisy word sequences is also important. In this paper, we focus on the NER of ASR results and discuss the suppression of ASR error problems in NER.

Most previous studies of the NER of speech data used generative models such as hidden Markov models (HMMs) (Miller et al., 1999; Palmer and Ostendorf, 2001; Horlock and King, 2003b; Béchet et al., 2004; Favre et al., 2005). On the other hand, in text-based NER, better results are obtained using discriminative schemes such as maximum entropy (ME) models (Borthwick, 1999; Chieu and Ng, 2003), support vector machines (SVMs) (Isozaki and Kazawa, 2002), and conditional random fields (CRFs) (McCallum and Li, 2003). Zhai et al. (2004) applied a text-level ME-based NER to ASR results. These models have an advantage in utilizing various features, such as part-of-speech information, character types, and surrounding words, which may be overlapped, while overlapping features are hard to use in HMM-based models.

To deal with ASR error problems in NER, Palmer and Ostendorf (2001) proposed an HMM-based NER method that explicitly models ASR errors using ASR confidence and rejects erroneous word hypotheses in the ASR results. Such rejection is especially effective when ASR accuracy is relatively low because many misrecognized words may be extracted as NEs, which would decrease NER precision.

Motivated by these issues, we extended their approach to discriminative models and propose an NER method that deals with ASR errors as fea-

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 617–624,
Sydney, July 2006. ©2006 Association for Computational Linguistics

tures. We use NE-labeled ASR results for training to incorporate the features into the NER model as well as the corresponding transcriptions with NE labels. In testing, ASR errors are identified by ASR confidence scores and are used for the NER. In experiments using SVM-based NER and speech data from Japanese newspaper articles, the proposed method 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 kind of chunking problem that can be solved by classifying words into NE classes that consist of name categories and such chunking states as `PERSON-BEGIN` (the beginning of a person's name) and `LOCATION-MIDDLE` (the middle of a location's name). Many discriminative methods have been applied to NER, such as decision trees (Sekine et al., 1998), ME models (Borthwick, 1999; Chieu and Ng, 2003), and CRFs (McCallum and Li, 2003). In this paper, we employ an SVM-based NER method in the following way that showed good NER performance in Japanese (Isozaki and Kazawa, 2002).

We define three features for each word: the word itself, its part-of-speech tag, and its character type. We also use those features for the two preceding and succeeding words for context dependence and use 15 features when classifying a word. Each feature is represented by a binary value (1 or 0), for example, "whether the previous word is *Japan*," and each word is classified based on a long binary vector where only 15 elements are 1.

We have two problems when solving NER using SVMs. One, SVMs can solve only a two-class problem. We reduce multi-class problems of NER to a group of two-class problems using the *one-against-all* approach, where each SVM is trained to distinguish members of a class (e.g., `PERSON-BEGIN`) from non-members (`PERSON-MIDDLE`, `MONEY-BEGIN`, ...). 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 is that the NE label sequence must be consistent; for example, `ARTIFACT-END` must follow `ARTIFACT-BEGIN` or `ARTIFACT-MIDDLE`. We use a Viterbi search to obtain the best and consistent NE label sequence after classifying all words in a sentence, based on probability-like values obtained by applying sigmoid function $s_n(x) = 1/(1 + \exp(-\beta_n x))$ to SVM output score $g_c(x)$.

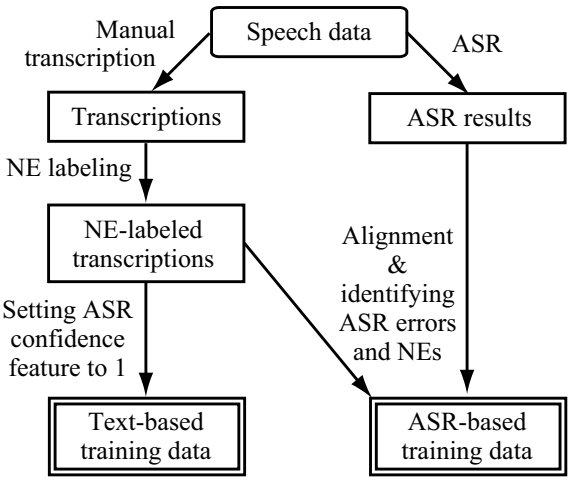

Figure 1: Procedure for preparing training data.

3 Proposed method

3.1 Incorporating ASR confidence into NER

In the NER of ASR results, ASR errors cause NEs to be missed and erroneous NEs to be recognized. If one or more words constituting an NE are misrecognized, we cannot recognize the correct NE. Even if all words constituting an NE are correctly recognized, we may not recognize the correct NE due to ASR errors on context words. To avoid this problem, we model ASR errors using additional features that indicate whether each word is correctly recognized. Our NER model is trained using ASR results with a feature, where feature values are obtained through alignment to the corresponding transcriptions. In testing, we estimate feature values using ASR confidence scores. In this paper, this feature is called the *ASR confidence feature*.

Note that we only aim to identify NEs that are correctly recognized by ASR, and NEs containing ASR errors are not regarded as NEs. Utilizing erroneous NEs is a more difficult problem that is beyond the scope of this paper.

3.2 Training NER model

Figure 1 illustrates the procedure for preparing training data from speech data. First, the speech

data are manually transcribed and automatically recognized by the ASR. Second, we label NEs in the transcriptions and then set the ASR confidence feature values to 1 because the words in the transcriptions are regarded as correctly recognized words. Finally, we align the ASR results to the transcriptions to identify ASR errors for the ASR confidence feature values and to label correctly recognized NEs in the ASR results. Note that we label the NEs in the ASR results that exist in the same positions as the transcriptions. If a part of an NE is misrecognized, the NE is ignored, and all words for the NE are labeled as non-NE words (`OTHER`). Examples of text-based and ASR-based training data are shown in Tables 1 and 2. Since the name *Murayama Tomiichi* in Table 1 is misrecognized in ASR, the correctly recognized word *Murayama* is also labeled `OTHER` in Table 2. Another approach can be considered, where misrecognized words are replaced by word error symbols such as those shown in Table 3. In this case, those words are rejected, and those part-of-speech and character type features are not used in NER.

3.3 ASR confidence scoring for using the proposed NER model

ASR confidence scoring is an important technique in many ASR applications, and many methods have been proposed including using word posterior probabilities on word graphs (Wessel et al., 2001), integrating several confidence measures using neural networks (Schaaf and Kemp, 1997), using linear discriminant analysis (Kamppari and Hazen, 2000), and using SVMs (Zhang and Rudnicky, 2001).

Word posterior probability is a commonly used and effective ASR confidence measure. Word posterior probability $p([w;\tau,t]|\boldsymbol{X})$ of word w at time interval $[\tau,t]$ for speech signal $\boldsymbol{X}$ is calculated as follows (Wessel et al., 2001):

$$p([w;\tau,t]|\boldsymbol{X}) = \sum_{\boldsymbol{W}\in\boldsymbol{W}[w;\tau,t]} \frac{\left\{p(\boldsymbol{X}|\boldsymbol{W})\,(p(\boldsymbol{W}))^{\beta}\right\}^{\alpha}}{p(\boldsymbol{X})}, \quad (1)$$

where $\boldsymbol{W}$ is a sentence hypothesis, $\boldsymbol{W}[w;\tau,t]$ is the set of sentence hypotheses that include w in $[\tau,t]$, $p(\boldsymbol{X}|\boldsymbol{W})$ is a acoustic model score, $p(\boldsymbol{W})$ is a language model score, α is a scaling parameter (α<1), and β is a language model weight. α is used for scaling the large dynamic range of $p(\boldsymbol{X}|\boldsymbol{W})(p(\boldsymbol{W}))^{\beta}$ to avoid a few of the top hypotheses dominating posterior probabilities. $p(\boldsymbol{X})$ is approximated by the sum over all sentence hypotheses and is denoted as

$$p(\boldsymbol{X}) = \sum_{\boldsymbol{W}} \left\{p(\boldsymbol{X}|\boldsymbol{W})\,(p(\boldsymbol{W}))^{\beta}\right\}^{\alpha}. \quad (2)$$

$p([w;\tau,t]|\boldsymbol{X})$ can be efficiently calculated using a forward-backward algorithm.

In this paper, we use SVMs for ASR confidence scoring to achieve a better performance than when using word posterior probabilities as ASR confidence scores. SVMs are trained using ASR results, whose errors are known through their alignment to their reference transcriptions. The following features are used for confidence scoring: the word itself, its part-of-speech tag, and its word posterior probability; those of the two preceding and succeeding words are also used. The word itself and its part-of-speech are also represented

Word	Confidence	NE label
Murayama	1	`PERSON-BEGIN`
Tomiichi	1	`PERSON-END`
shusho	1	`OTHER`
wa	1	`OTHER`
nento	1	`DATE-SINGLE`

Table 1: An example of text-based training data.

Word	Confidence	NE label
Murayama	1	`OTHER`
shi	0	`OTHER`
ni	0	`OTHER`
ichi	0	`OTHER`
shiyo	0	`OTHER`
wa	1	`OTHER`
nento	1	`DATE-SINGLE`

Table 2: An example of ASR-based training data.

Word	Confidence	NE label
Murayama	1	`OTHER`
(error)	0	`OTHER`
(error)	0	`OTHER`
(error)	0	`OTHER`
(error)	0	`OTHER`
wa	1	`OTHER`
nento	1	`DATE-SINGLE`

Table 3: An example of ASR-based training data with word error symbols.

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 for simplicity: (if $0 < p \leq 0.1$, if $0.1 < p \leq 0.2$, ... , and if $0.9 < p \leq 1.0$). To normalize SVMs' output scores for ASR confidence, we use a sigmoid function $s_w(x) = 1/(1 + \exp(-\beta_w x))$. We use these normalized scores as ASR confidence scores. Although a large variety of features have been proposed in previous studies, we use only these simple features and reserve the other features for further studies.

Using the ASR confidence scores, we estimate whether each word is correctly recognized. If the ASR confidence score of a word is greater than threshold t_w, the word is estimated as correct, and we set the ASR confidence feature value to 1; otherwise we set it to 0.

3.4 Rejection at the NER level

We use the ASR confidence feature to suppress ASR error problems; however, even text-based NERs sometimes make errors. NER performance is a trade-off between missing correct NEs and accepting erroneous NEs, and requirements differ by task. Although we can tune the parameters in training SVMs to control the trade-off, it seems very hard to find appropriate values for all the SVMs. We use a simple NER-level rejection by modifying the SVM output scores for the non-NE class (`OTHER`). We add constant offset value t_o to each SVM output score for `OTHER`. With a large t_o, `OTHER` becomes more desirable than the other NE classes, and many words are classified as non-NE words and *vice versa*. Therefore, t_o works as a parameter for NER-level rejection. This approach can also be applied to text-based NER.

4 Experiments

We conducted the following experiments related to the NER of speech data to investigate the performance of the proposed method.

4.1 Setup

In the experiment, we simulated the procedure shown in Figure 1 using speech data from the NE-labeled text corpus. We used the training data of the Information Retrieval and Extraction Exercise (IREX) workshop (Sekine and Eriguchi, 2000) as the text corpus, which consisted of 1,174 Japanese newspaper articles (10,718 sentences) and 18,200 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 corresponding 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 [1] 2.3.3 and then labeled the NEs. Unreadable tokens such as parentheses were removed in tokenization. After tokenization, the text corpus had 264,388 words of 60 part-of-speech types. Since three different kinds of characters are used in Japanese, the character types used as features included: `single-kanji` (words written in 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 was performed). The chunking states included in the NE classes were: `BEGIN` (beginning of a NE), `MIDDLE` (middle of a NE), `END` (ending of a NE), and `SINGLE` (a single-word NE). There were 33 NE classes (eight categories * four chunking states + `OTHER`), and therefore we trained 33 SVMs to distinguish words of a class from words of other classes. For NER, we used an SVM-based chunk annotator YamCha [2] 0.33 with a quadratic kernel $(1 + \vec{\boldsymbol{x}} \cdot \vec{\boldsymbol{y}})^2$ and a soft margin parameter of SVMs C=0.1 for training and applied sigmoid function $s_n(x)$ with β_n=1.0 and Viterbi search to the SVMs' outputs. These parameters were experimentally chosen using the test set.

We used an ASR engine (Hori et al., 2004) with a speaker-independent acoustic model. The lan-

[1] http://chasen.naist.jp/hiki/ChaSen/ (in Japanese)
[2] http://www.chasen.org/~taku/software/yamcha/

guage model was a word 3-gram model, trained using other Japanese newspaper articles (about 340 M words) that were also tokenized using ChaSen. The vocabulary size of the word 3-gram model was 426,023. The test-set perplexity over the text corpus was 76.928. The number of out-of-vocabulary words was 1,551 (0.587%). 223 (1.23%) NEs in the text corpus contained such out-of-vocabulary words, so those NEs could not be correctly recognized by ASR. The scaling parameter α was set to 0.01, which showed the best ASR error estimation results using word posterior probabilities in the test set in terms of receiver operator characteristic (ROC) curves. The language model weight β was set to 15, which is a commonly used value in our ASR system. The word accuracy obtained using our ASR engine for the overall dataset was 79.45%. In the ASR results, 82.00% of the NEs in the text corpus remained. Figure 2 shows the ROC curves of ASR error estimation for the overall five cross-validation test sets, using SVM-based ASR confidence scoring and word posterior probabilities as ASR confidence scores, where

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

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

In SVM-based ASR confidence scoring, we used the quadratic kernel and C=0.01. Parameter β_w of sigmoid function $s_w(x)$ was set to 1.0. These parameters were also experimentally chosen. SVM-based ASR confidence scoring showed better performance in ASR error estimation than simple word posterior probabilities by integrating multiple features. Five values of ASR confidence threshold t_w were tested in the following experiments: 0.2, 0.3, 0.4, 0.5, and 0.6 (shown by black dots in Figure 2).

4.2 Evaluation metrics

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

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

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

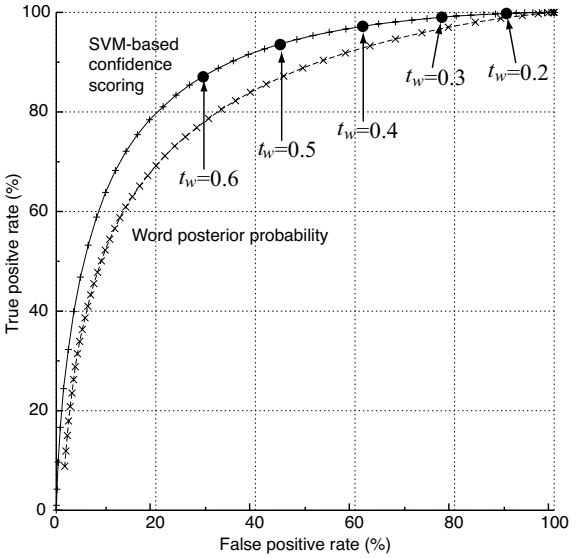

Figure 2: SVM-based confidence scoring outperforms word posterior probability for ASR error estimation.

A recognized NE was accepted as correct if and only if it appeared in the same position as its reference NE through alignment, in addition to having the correct NE surface and category, because the same NEs might appear more than once. Comparisons of NE surfaces did not include differences in word segmentation because of the segmentation ambiguity in Japanese. Note that NER recall with ASR results could not exceed the rate of the remaining NEs after ASR (about 82%) because NEs containing ASR errors were always lost.

In addition, we also evaluated the NER performance in NER precision and recall with NER-level rejection using the procedure in Section 3.4, by modifying the non-NE class scores using offset value t_o.

4.3 Compared methods

We compared several combinations of features and training conditions for evaluating the effect of incorporating the ASR confidence feature and investigating differences among training data: text-based, ASR-based, and both.

Baseline does not use the ASR confidence feature and is trained using text-based training data only.

NoConf-A does not use the ASR confidence feature and is trained using ASR-based training data only.

Method	Confidence	Training	Test	F-measure (%)	Precision (%)	Recall (%)
Baseline	Not used	Text	ASR	67.00	70.67	63.70
NoConf-A		ASR	ASR	65.52	78.86	56.05
NoConf-TA		Text+ASR	ASR	66.95	77.55	58.91
Conf-A	Used	ASR	ASR*	67.69	76.69	60.59
Proposed		Text+ASR	ASR*	**69.02**	**78.13**	**61.81**
Conf-Reject	Used†	Text+ASR	ASR*	68.77	77.57	61.78
Conf-UB	Used	Text+ASR	ASR**	73.14	87.51	62.83
Transcription	Not used	Text	Text	84.04	86.27	81.93

Table 4: NER results in averaged NER F-measure, precision, and recall without considering NER-level rejection ($t_o = 0$). ASR word accuracy was 79.45%, and 82.00% of NEs remained in ASR results. (†*Unconfident words were rejected and replaced by word error symbols,* $^{*}t_w = 0.4$, ***ASR errors were known.*)

NoConf-TA does not use the ASR confidence feature and is trained using both text-based and ASR-based training data.

Conf-A uses the ASR confidence feature and is trained using ASR-based training data only.

Proposed uses the ASR confidence feature and is trained using both text-based and ASR-based training data.

Conf-Reject is almost the same as Proposed, but misrecognized words are rejected and replaced with word error symbols, as described at the end of Section 3.2.

The following two methods are for reference.

Conf-UB assumes perfect ASR confidence scoring, so the ASR errors in the test set are known. The NER model, which is identical to Proposed, is regarded as the upper-boundary of Proposed.

Transcription applies the same model as Baseline to reference transcriptions, assuming word accuracy is 100%.

4.4 NER Results

In the NER experiments, Proposed achieved the best results among the above methods. Table 4 shows the NER results obtained by the methods without considering NER-level rejection (i.e., $t_o = 0$), using threshold $t_w = 0.4$ for Conf-A, Proposed, and Conf-Reject, which resulted in the best NER F-measures (see Table 5). Proposed showed the best F-measure, 69.02%. It outperformed Baseline by 2.0%, with a 7.5% improvement in precision, instead of a recall decrease of 1.9%. Conf-Reject showed slightly worse results than Proposed. Conf-A resulted in 1.3% worse F-measure than Proposed. NoConf-A and NoConf-TA achieved 7-8% higher precision than Baseline; however, their F-measure results were worse than Baseline because of the large drop of recall. The upper-bound results of the proposed method (Conf-UB) in F-measure was 73.14%, which was 4% higher than Proposed.

Method	t_w	F (%)	P (%)	R (%)
Conf-A	0.2	66.72	71.28	62.71
	0.3	67.32	73.68	61.98
	0.4	**67.69**	**76.69**	**60.59**
	0.5	67.04	79.64	57.89
	0.6	64.48	81.90	53.14
Proposed	0.2	68.08	72.54	64.14
	0.3	68.70	75.11	63.31
	0.4	**69.02**	**78.13**	**61.81**
	0.5	68.17	80.88	58.93
	0.6	65.39	83.00	53.96
Conf-Reject	0.2	68.06	72.49	64.14
	0.3	68.61	74.88	63.31
	0.4	**68.77**	**77.57**	**61.78**
	0.5	67.93	80.23	58.91
	0.6	64.93	82.05	53.73

Table 5: NER results with varying ASR confidence score threshold t_w for Conf-A, Proposed, and Conf-Reject. (F: F-measure, P: precision, R: recall)

Figure 3 shows NER precision and recall with NER-level rejection by t_o for Baseline, NoConf-TA, Proposed, Conf-UB, and Transcription. In the figure, black dots represent results with $t_o = 0$, as shown in Table 4. By all five methods, we

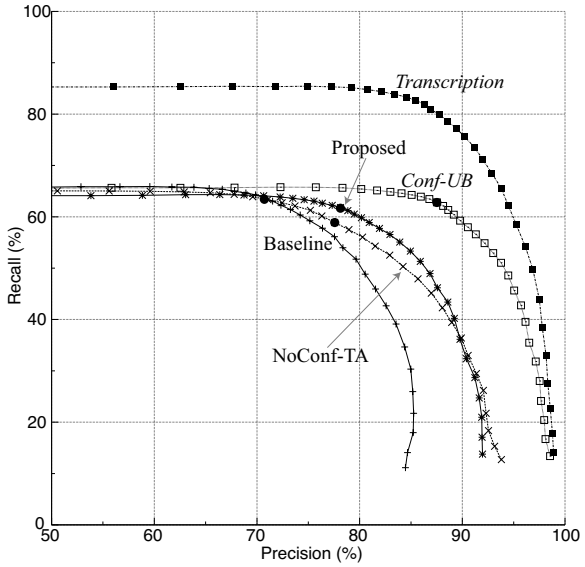

Figure 3: NER precision and recall with NER-level rejection by t_o

obtained higher precision with $t_o > 0$. Proposed achieved more than 5% higher precision than Baseline on most recall ranges and showed higher precision than NoConf-TA on recall ranges higher than about 35%.

5 Discussion

The proposed method effectively improves NER performance, as shown by the difference between Proposed and Baseline in Tables 4 and 5. Improvement comes from two factors: using both text-based and ASR-based training data and incorporating ASR confidence feature. As shown by the difference between Baseline and the methods using ASR-based training data (NoConf-A, NoConf-TA, Conf-A, Proposed, Conf-Reject), ASR-based training data increases precision and decreases recall. In ASR-based training data, all words constituting NEs that contain ASR errors are regarded as non-NE words, and those NE examples are lost in training, which emphasizes NER precision. When text-based training data are also available, they compensate for the loss of NE examples and recover NER recall, as shown by the difference between the methods without text-based training data (NoConf-A, Conf-A) and those with (NoConf-TA, Proposed). The ASR confidence feature also increases NER recall, as shown by the difference between the methods without it (NoConf-A, NoConf-TA) and with it (Conf-A, Proposed). This suggests that the ASR confidence feature helps distinguish whether ASR error influences NER and suppresses excessive rejection of NEs around ASR errors.

With respect to the ASR confidence feature, the small difference between Conf-Reject and Proposed suggests that ASR confidence is a more dominant feature in misrecognized words than the other features: the word itself, its part-of-speech tag, and its character type. In addition, the difference between Conf-UB and Proposed indicated that there is room to improve NER performance with better ASR confidence scoring.

NER-level rejection also increased precision, as shown in Figure 3. We can control the trade-off between precision and recall with t_o according to the task requirements, even in text-based NER. In the NER of speech data, we can obtain much higher precision using both ASR-based training data and NER-level rejection than using either one.

6 Related work

Recent studies on the NER of speech data consider more than 1-best ASR results in the form of N-best lists and word lattices. Using many ASR hypotheses helps recover the ASR errors of NE words in 1-best ASR results and improves NER accuracy. Our method can be extended to multiple ASR hypotheses.

Generative NER models were used for multi-pass ASR and NER searches using word lattices (Horlock and King, 2003b; Béchet et al., 2004; Favre et al., 2005). Horlock and King (2003a) also proposed discriminative training of their NER models. These studies showed the advantage of using multiple ASR hypotheses, but they do not use overlapping features.

Discriminative NER models were also applied to multiple ASR hypotheses. Zhai et al. (2004) applied text-based NER to N-best ASR results, and merged the N-best NER results by weighted voting based on several sentence-level results such as ASR and NER scores. Using the ASR confidence feature does not depend on SVMs and can be used with their method and other discriminative models.

7 Conclusion

We proposed a method for NER of speech data that incorporates ASR confidence as a feature of discriminative NER, where the NER model

is trained using both text-based and ASR-based training data. In experiments using SVMs, the proposed method showed a higher NER F-measure, especially in terms of improving precision, than simply applying text-based NER to ASR results. The method effectively rejected erroneous NEs due to ASR errors with a small drop of recall, thanks to both the ASR confidence feature and ASR-based training data. NER-level rejection also effectively increased precision.

Our approach can also be used in other tasks in spoken language processing, and we expect it to be effective. Since confidence itself is not limited to speech, our approach can also be applied to other noisy inputs, such as optical character recognition (OCR). For further improvement, we will consider N-best ASR results or word lattices as inputs and introduce more speech-specific features such as word durations and prosodic features.

Acknowledgments We would like to thank anonymous reviewers for their helpful comments.

References

Frédéric Béchet, Allen L. Gorin, Jeremy H. Wright, and Dilek Hakkani-Tür. 2004. Detecting and extracting named entities from spontaneous speech in a mixed-initiative spoken dialogue context: How May I Help You? *Speech Communication*, 42(2):207–225.

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

Hai Leong Chieu and Hwee Tou Ng. 2003. Named entity recognition with a maximum entropy approach. In *Proc. CoNLL*, pages 160–163.

Benoît Favre, Frédéric Béchet, and Pascal Nocéra. 2005. Robust named entity extraction from large spoken archives. In *Proc. HLT-EMNLP*, pages 491–498.

Takaaki Hori, Chiori Hori, and Yasuhiro Minami. 2004. Fast on-the-fly composition for weighted finite-state transducers in 1.8 million-word vocabulary continuous-speech recognition. In *Proc. ICSLP*, volume 1, pages 289–292.

James Horlock and Simon King. 2003a. Discriminative methods for improving named entity extraction on speech data. In *Proc. EUROSPEECH*, pages 2765–2768.

James Horlock and Simon King. 2003b. Named entity extraction from word lattices. In *Proc. EUROSPEECH*, pages 1265–1268.

Hideki Isozaki and Hideto Kazawa. 2002. Efficient support vector classifiers for named entity recognition. In *Proc. COLING*, pages 390–396.

Simo O. Kamppari and Timothy J. Hazen. 2000. Word and phone level acoustic confidence scoring. In *Proc. ICASSP*, volume 3, pages 1799–1802.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proc. CoNLL*, pages 188–191.

David Miller, Richard Schwartz, Ralph Weischedel, and Rebecca Stone. 1999. Named entity extraction from broadcast news. In *Proceedings of the DARPA Broadcast News Workshop*, pages 37–40.

David D. Palmer and Mari Ostendorf. 2001. Improving information extraction by modeling errors in speech recognizer output. In *Proc. HLT*, pages 156–160.

Thomas Schaaf and Thomas Kemp. 1997. Confidence measures for spontaneous speech recognition. In *Proc. ICASSP*, volume II, pages 875–878.

Satoshi Sekine and Yoshio Eriguchi. 2000. Japanese named entity extraction evaluation - analysis of results. In *Proc. COLING*, pages 25–30.

Satoshi Sekine, Ralph Grishman, and Hiroyuki Shinnou. 1998. A decision tree method for finding and classifying names in Japanese texts. In *Proc. the Sixth Workshop on Very Large Corpora*, pages 171–178.

Frank Wessel, Ralf Schlüter, Klaus Macherey, and Hermann Ney. 2001. Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 9(3):288–298.

Lufeng Zhai, Pascale Fung, Richard Schwartz, Marine Carpuat, and Dekai Wu. 2004. Using N-best lists for named entity recognition from chinese speech. In *Proc. HLT-NAACL*, pages 37–40.

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