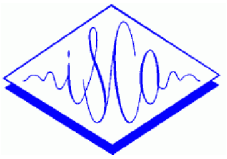

International Workshop on Spoken
Language Translation (IWSLT) 2005
Pittsburgh, PA, USA
October 24-25, 2005

A Decoding Algorithm for Word Lattice Translation in Speech Translation

Ruiqiang Zhang, Genichiro Kikui, Hirofumi Yamamoto, Wai-Kit Lo

ATR Spoken Language Translation Research Laboratories
2-2 Hikaridai, Seiika-cho, Soraku-gun, Kyoto, 619-0288, Japan
{ruiqiang.zhang, genichiro.kikui, hirofumi.yamamoto,waikit.lo}@atr.jp

Abstract

We propose a novel statistical machine translation decoding algorithm for speech translation to improve speech translation quality. The algorithm can translate the speech recognition word lattice, where more hypotheses are utilized to bypass the misrecognized single-best hypotheses. We also show that a speech recognition confidence measure, implemented by posterior probability, is effective to improve speech translation. The proposed techniques were tested in a Japanese-to-English speech translation task. The experimental results demonstrate the improved speech translation performance by the proposed techniques.

1. Introduction

Most current speech translation systems have a cascaded structure: a speech recognition component followed by a machine translation component. Usually only the single-best outcome of speech recognition is used in the machine translation component. Due to the inevitable errors of speech recognition, speech translation cannot achieve the same level of translation performance as that achieved by perfect text input.

To overcome the weakness in speech translation, several architectures have been proposed so far. [1] proposed a coupling structure to combine automatic speech recognition and statistical machine translation. [2] used a unified structure where the maximum entropy approach is proposed to build entire speech translation system models. [3] and [4] implemented the integrated structure by means of finite-state network, and a comparison with the cascaded structure was made. Strictly speaking, this approach does not use the statistical speech translation structure [1].

In this work we used the speech recognition word lattice as the output of speech recognition and the input of machine translation. While in this structure the speech recognition component and the machine translation component are sequentially connected, more hypotheses are stored in the word lattice than the single-best structure; complementary information, such as the acoustic model and language model scores instantiated by posterior probability, was used in the machine translation component to enhance translation performance.

In the field of statistical machine translation, the famous, early models are the IBM models [5], using Bayes rule to convert $P(e|f)$ into $P(e)P(f|e)$. The IBM Models 1 through 5 introduced various models for $P(f|e)$ with increasing complexity. Another popular model is called an "HMM" model [6], which adds alignment probability in the basis of IBM Model 1. Recently a direct modeling of $P(e|f)$ in the maximum entropy framework, the log-linear model, has been proved effective [7]. This model can integrate a number of features log-linearly. Hence, we use the statistical log-linear model as the translation model in this work.

We implemented a new lattice translation decoding algorithm specialized for speech translation. In the decoding we used a two-pass search strategy, graph-based plus A*. For the first graph search, we integrated features from IBM Model 1 into the log-linear model while IBM Model 4's features were integrated in the second A* search. We invented a new method to minimize the size of the raw lattice generated by the speech recognizer to reduce the decoding complexity. We found these techniques effective for improving speech translation quality.

We also found that sentence posterior probability of speech recognition was very useful for further improving speech translation. A significant translation improvement was achieved by filtering low-confidence hypotheses based on the posterior probability in the word lattice.

The remaining sections are organized as follows. Section 2 introduces our speech translation models and structure, then Section 3 provides detailed descriptions of the decoding algorithm of the word lattice translation. In Section 4 we describe a lattice reduction method to reduce the computations. We describe the sentence posterior probability approach for choosing hypotheses in Section 5. Section 6 presents our experimental results and a detailed analysis, and Section 7 gives our discussions and conclusions.

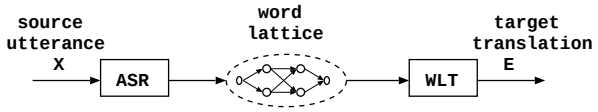

Figure 1: Speech Translation Framework

2. Proposed Speech Translation Structure

The proposed speech translation system is illustrated in Figure 1. It consists of two major components: an automatic speech recognition (ASR) module and a word lattice translation (WLT) module. The interface between the two components is a recognition word lattice.

The task of speech translation, in the case of Japanese-to-English translation, can be modeled as to find the target English sentence, $\widehat{E}$, given a source Japanese spoken utterance, X, such that the probability, $P(E|X)$, is maximized. If the intermediate output of ASR is defined as J, we can get the following formula according to [4]:

$$\begin{aligned}\widehat{E} &= \arg\max_{E} P(E|X) \\ &= \arg\max_{E} \left\{ \sum_{J} P(J,E)P(X|J) \right\} \end{aligned} \quad (1)$$

where $P(X|J)$ is the ASR acoustic model; $P(J,E)$, the joint model of source and target languages.

For searching the best E, two methods are used in [4]: serial structure and integrated stochastic finite-state transducer(SFST). The first used the single-best results of ASR, and the second used a finite-state transducer.

We use the word lattice in our work, by which the implementation indicated by the model 1 is approximated in two steps:

- First, use the ASR component to generate a word lattice G. Only the top ranked hypotheses with an ASR scores higher than a threshold, TH, are kept in the word lattice:

$$G = \{J | P(J)P(X|J) > TH\} \quad (2)$$

- Second, use the WLT component to find output which maximizes:

$$< \widehat{E}, \widehat{J} > = \arg\max_{E,J} \{P(E)P(X|J)P(J|E)\} \quad J \in G \quad (3)$$

The combination of Eq. 2 and Eq. 3 is an approximation of Eq. 1, where we assume E and X is independent to derive Eq. 2; and summation is replaced by maximization to derive Eq. 3. Meanwhile, a $\widehat{J}$, producing the best $\widehat{E}$, is also obtained in WLT despite that the goal of speech translation is only for $\widehat{E}$.

Although the WLT translation model is derived in the form of Eq. 3, we actually used a more advanced model, namely a feature based log-linear models, formalized as:

$$\begin{aligned}\widehat{E} &= \arg\max_{E} \{\lambda_0 \log P_{pp}(J|X) + \lambda_1 \log P_{lm}(E) \\ &+ \lambda_2 \log P_{lm}(\mathcal{POS}(E)) + \lambda_3 \log \mathcal{N}(\Phi|E) \\ &+ \lambda_4 \log P(\Phi_0|E) + \lambda_5 \log \mathcal{T}(J|E) \\ &+ \lambda_6 \log \mathcal{D}(E,J)\} \end{aligned} \quad (4)$$

where we defined seven features represented by the following.

(a)ASR hypothesis posterior probability, $P_{pp}(J|X)$. We used the posterior probability instead of the acoustic model score since the acoustic model score has a large dynamic range and difficult to normalize. The posterior probability is calculated as:

$$P(J|X) = \frac{P(X|J)P(J)}{\sum_{J_i} P(X|J_i)P(J_i)} \quad (5)$$

where the summation is made over all hypotheses in the word lattice, G. J_i is a hypothesis in the word lattice. (b)Word sequence target language model, $P_{lm}(E)$. (c)Part-of-speech sequence target language models, $P_{lm}(\mathcal{POS}(E))$. (d)Fertility model, $\mathcal{N}(\Phi|E)$ represents the probability of the English word, e, generating ϕ words. (e)NULL translation model, $P(\Phi_0|E)$ is the probability of inserting a NULL word. (f)Lexicon Model, $\mathcal{T}(J|E)$ is the probability of the word, j, in the Japanese source sentence being translated into the corresponding word, e, in the English target sentence. (g)Distortion model, $\mathcal{D}(E,J)$ indicates the alignment probability of the source and target sentence,(J,E).

Eq. 4 is a logarithmic extension of Eq. 1 except that the translation model $P(J|E)$ is extended by IBM model 4 [5] and the acoustic feature is replaced by the posterior probability.

3. Word lattice translation – WLT

Word lattice translation is much more complicated than text translation. In contrast to text translation where a single source sentence is known, there is no single source sentence for word lattice translation but a lattice containing multiple hypotheses. Which hypothesis is the best one to be translated is unknown before the decoding is completed.

We use the graph+A* decoding approach for the word lattice translation. This approach has been used for text translation by [8]. We extend the approach to speech translation in this work. We adopted this approach because it can keep more hypotheses in a compact structure. The graph+A* decoding is a two-pass decoding. The first pass uses a simple model to generate a word graph to save the most likely hypotheses. It amounts to

converting a source language word lattice (SWL) into a target language word graph (TWG). Edges in the SWL are aligned to some edges in the TWG. The second pass uses a complicated model to output the best hypothesis by traversing the target word graph.

We describe the two-pass WLT algorithm in the following two sections.

3.1. First pass — from SWL to TWG

The bottom in Fig. 2 shows an example of a translation word graph, which corresponds to the recognition word lattice in the top. Each edge in the TWG is a target language word being a translation of a source word in the SWL, either from word translations provided by the lexical models or fertility expansion by the fertility models. Some edges that have the same structure are merged into a node. The node has one element indicating the source word coverage up to the current node. The coverage is a binary vector with size equal to the number of edges in the SWL, indicating the number of translated source edges. If the j-th source word was translated, the j-th element is 1, otherwise it equals to 0. If the node covers all the edges of a full path in the SWL, this node connects to the last node, the end, in the TWG.

There are two main operations in expanding a node into edges: DIRECT and ALIGN. DIRECT extends the hypothesis with a target word by translating an uncovered source word. The target word is determined based on current target N-gram context and possible translations of the uncovered source word.

ALIGN extends the hypothesis by aligning one more uncovered source word to the current node, where the target word is a translation of multiple source words, increasing fertilities of the target word.

The edge is not extended if the resulted hypothesis does not correspond to any hypothesis in the SWL. If the node has covered a full path in the SWL, this node is connected to the end node. When there are no nodes available for possible extension, the conversion is completed. A simple illustration of conversion algorithm is shown in Algorithm 1. The whole process equals the growing of a graph. The graph can be indexed in time slices because the new nodes are created based on the old nodes of the last nearest time slice. New nodes are created by DIRECT or ALIGN to cover the uncovered source edge and connect to the old nodes. The new generated nodes are sorted and merged in the graph buffer if they share the same structure: the same coverage, the same translations and the same N-gram sequence. If the node covers a full hypothesis in the SWL, the node connects to the end. If no nodes need to be expanded, the conversion finishes.

In the first pass, we incorporate a simpler translation model into the log-linear model, only the lexical model, IBM model 1. The ASR posterior probability P_{pp} are calculated by partial hypothesis from the start to the current.

Algorithm 1 Conversion Algorithm from SWL to TWG

```
1: Initialize graph buffer G[0]=0; t=0
2: DO
3:     FOR EACH node n=0,1,..., #(G[t]) DO
4:     IF (n cover A FULL PATH) NEXT
5:         FOR EACH edge l=0,1,...,#(EDGES) DO
6:             IF (n cover l) NEXT
7:             IF (n not cover ANY SWL PATH) NEXT
8:             generate new node and push to G[t+1]
9:         merge and prune nodes in G[t+1]
10:    t= t+1
11:WHILE (G[t] is empty)
```

P_{pp} uses the highest value among all the ASR hypotheses under the current context. The first pass serves to keep the most likely hypotheses in the translation word graph, and the second pass as a refined search that uses advanced models.

3.2. Second pass — by an A* search to find the best output from the TWG

An A* search traverses the TWG generated in last section, and this is the best first approach. All partial hypotheses generated are pushed into a priority queue with the top hypothesis popping first out of the queue for the next extension.

To execute the A* search, the hypothesis score, $D(h, n)$, of a node n is evaluated in two parts: the forward score, $F(h, n)$, and the heuristic estimation, $H(h, n)$, $D(h, n) = F(h, n) + H(h, n)$. The calculation of $F(h, n)$ begins from the start node and accumulates all nodes' scores belonging to the hypothesis until the current node, n. The $H(h, n)$ is defined as the accumulated maximum probability of the models from the end node to the current node n.

In the second pass we incorporated IBM Model 4's features into the log-linear model. However, we cannot use IBM Model 4 directly because the calculations of the two models, $P(\Phi_0|E)$ and $\mathcal{D}(E, J)$, require the source sentence, but unknown in fact. Hence, the probability of $P(\Phi_0|E)$ and $\mathcal{D}(E, J)$ cannot be calculated precisely in decoding. Our method to fix this problem is to use the maximum over all possible hypotheses. For the above two models, we calculated the scores for all the possible ASR hypotheses under the current context. The maximum value was used as the model's probability.

4. Minimizing the SWL

Because we use HMM-based ASR to generate the raw SWL, the same word identity can be recognized repeatedly in slightly different frames. As a result, the same

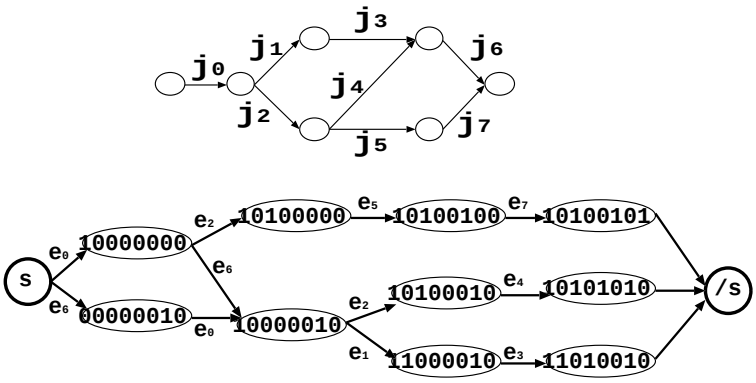

Figure 2: Source language word lattice (top) and target language word graph (bottom)

word identity may appear in more than one edge. Direct conversion from SWL to TWG causes duplicated computation and explosion of the TWG space. On the other hand, while the raw SWL contains hundreds of hypotheses, the top N-best hypotheses are among the most significant, which are only a small portion of all hypotheses. We can reduce the size of the raw SWL by cutting off all other hypotheses except the top N-best without information loss.

In reducing the size of SWL, we follow one rule: the TWG is the translation counterpart of the SWL if and only if any full path in the TWG is a translation of a full path in the SWL.

We use the following steps to downsize the raw SWL. From the raw SWL we generate N-best hypotheses in a sequence of edge numbers. We list the word IDs of all the edges in the hypotheses, remove the duplicate words, and index the remainders with new edge IDs. The number of new edges is less than that in the raw SWL. Next, we replace the edge sequence in each hypothesis with a new edge ID. If more than one edge shares the same word ID in one hypothesis, we add a new edge ID for the word again and replace the edge with the new ID. Finally, we generate a new word lattice with a new word list as its edges, consisting of the N-best hypotheses only. The raw SWL becomes the downsized SWL. The downsized SWL is much smaller than the raw SWL. In fact, in our experiments the word lattice is reduced by 50% on average.

Fig. 3 shows an example of lattice downsizing. The word IDs are shown in the parentheses. After downsizing, one hypothesis is removed. In fact, the downsized SWL is just the N-best ASR hypotheses with new assigned edge IDs. In this paper we use **lattice-hypothesis** to indicate the quantities of hypotheses in the lattice, defined as the number of hypotheses used to construct the TWG in the downsized SWL. It is a more suitable metric than lattice density for the downsized SWL because after lattice minimization, the downsized SWL saves only the significant N-best ASR hypotheses regardless of the density of the raw SWL.

5. Selection of hypotheses by confidence measure (CM) filtering

As described above, the downsized SWL stores N ASR hypotheses. All the hypotheses can find a counterpart in the TWG after the conversion if they are not removed by histogram and threshold pruning. The posterior probability of a hypothesis can determine whether this hypothesis is used in the TWG. In general, the hypotheses with the lowest posterior probability are the least likely to be used in the WLT module. They are most likely pruned in the earlier stage of the decoding process. In the experiments we found removing the hypotheses with extremely low posterior probability in the TWG by hard decisions can improve speech translation. We found that using posterior probability as a confidence measure to filter low confidence hypotheses achieved better results. For all the hypotheses in the SWL, we used Eq. 5 to compute each hypothesis posterior probability. We then compared each one's posterior probability with that of the single-best hypothesis, P_0, the highest posterior probability. If it exceeds a threshold, P_0/T, where T is a confidence factor, the hypothesis is used in WLT, otherwise removed. By applying a confidence measure, WLT automatically selects the hypotheses that are conversed into the TWG. Hence, for a given SWL, the number of hypotheses for translation in WLT is determined by the confidence measure (CM) filtering.

6. Experiments

6.1. BTEC Database & Model Training

The Japanese/English bilingual data used in this study were from the Basic Travel Expression Corpus (BTEC) [9], consisting of commonly used sentences published in travel guidebooks and tour conversations.

In our experiments we used the BTEC training data to

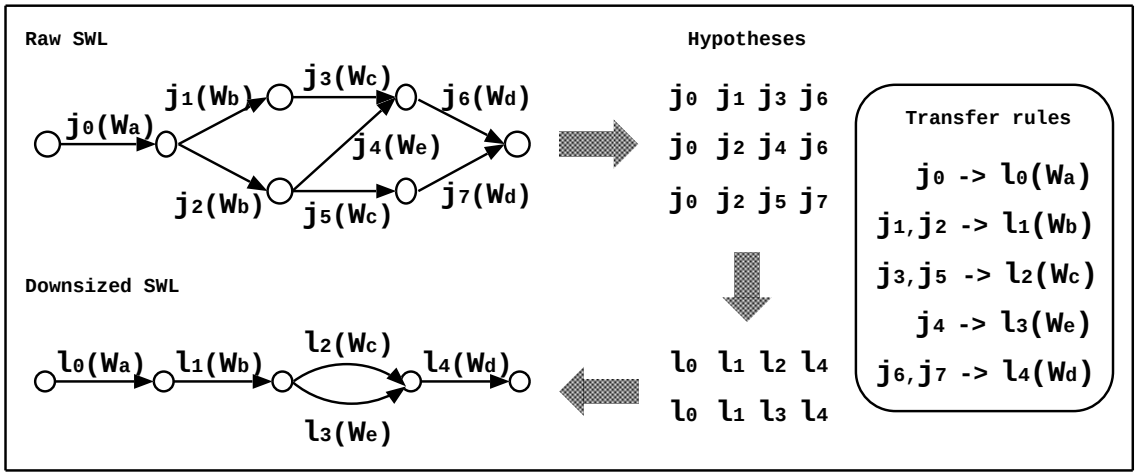

Figure 3: An example of word lattice reduction

train the models, the BTEC1 test data #1 as the development data for the parameter optimization of the log-linear model, and the BTEC1 test data #2 for evaluation. The training data contains 468,595 sentences. The develop and test data have 510 and 508 sentences respectively.

The adopted ASR is HMM-based, implemented by triphone models with 2,100 states and 25 dimensional MFCC features. A multiclass word bigram and a word trigram language models were used in the ASR with a lexicon of 47,000 words. All the LMs and the feature models of the translation models were trained by the BTEC training data.

The automatic evaluation metric, BLEU, was used for evaluating our translation quality. It was calculated by the downloadable tool (version v11a) [1]. We use 16 reference sentences for each utterance, created by human translators

In the development phase, we optimized the λs of the log-linear model by the approach in [7], for maximizing the BLEU score of the translation results of the development data given the 16 references for each utterance.

6.2. The effect of CM filtering

In the experiments the ASR system output the raw lattice. Its performance for one of our test data (BTEC test #2) is shown in Fig. 4, where the word and sentence accuracy for the single-best (lattice-hypothesis=1) recognition are around 93% and 79%, respectively. They increase to 96% and 87% at lattice-hypothesis=20, showing the potential improvement for speech translation.

The required lattice for WLT was generated by the lattice reduction approach described in section 4. We set the number of ASR hypotheses to 100 when we created the downsized SWL. The downsized SWL was then

[1] http://www.nist.gov/speech/tests/mt/

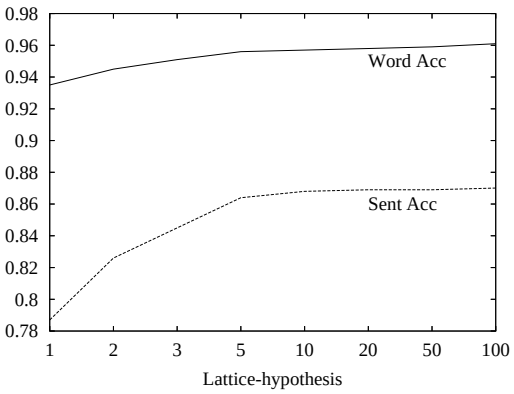

Figure 4: Speech recognition word accuracy and sentence accuracy

translated by the WLT translation module. While the downsized SWL contained 100 ASR hypotheses, the actual number of hypotheses used in WLT can be designated by changing the lattice-hypothesis. For a given lattice-hypothesis, we carried out the translation experiments under the conditions of with and without CM filtering. When did with CM filtering, the number of hypotheses used decreased further.

Figure 5 presents the translation results of WLT, showing the change of BLEU with increasing lattice-hypothesis. The bottom curve represents the translation without CM filtering while the top curve is the one with it. The results of lattice-hypothesis=1 corresponds to the single-best translation, the worst BLEU score. The BLEU scores of both translations increase as the lattice-hypothesis increases, and we found that the results when lattice-hypothesis=20 are the best for both. It proves the effectiveness of lattice translation in improving translation.

This experiments prove the speech translation with

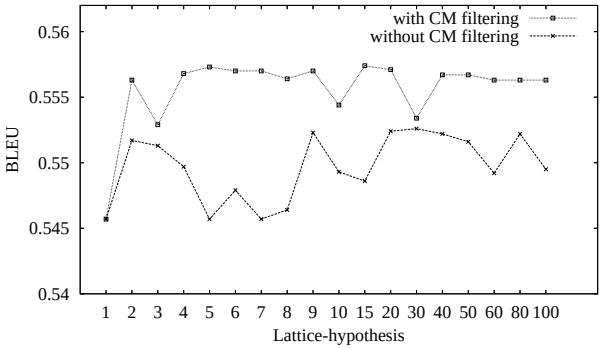

Figure 5: Effect of confidence measure filtering

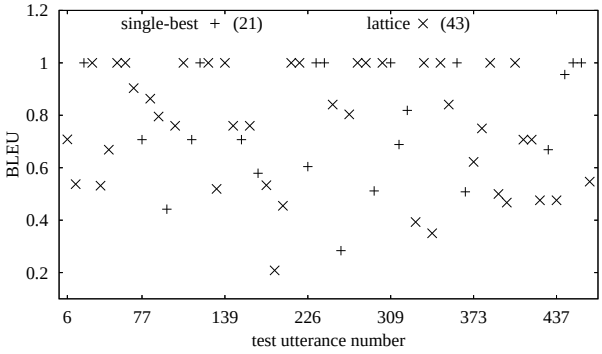

Figure 6: Contest of single-best and lattice translation: sign 'x' means lattice translation beats single-best; sign '+', opposite

CM filtering is more effective than that without CM filtering. The translation improvement with CM filtering is more stable when the lattice-hypothesis increases, and the improvement's amplitude is greater than that without CM filtering. On the contrary, the translation improvement without CM filtering is unstable and fluctuating, and in some case no better translations found than the single-best.

The results of Figure. 5 were obtained when the confidence threshold was set to $T = 10$, the best translation results observed.

6.3. The effect of word lattice translation

In the last section we showed the effect of CM filtering, proving that the CM filtering can improve lattice translation significantly in comparison with without CM filtering. As shown in the Fig. 5, the BLEU score was increased from the single-best translation, 0.545, to the lattice translation, 0.557. Although the overall improvement by lattice translation, indicated by the BLEU score, is sightly higher than the baseline single-best translation, this improvement is reliable because we measured the results by other popular evaluation metrics besides BLEU: NIST,mWER and mPER. We found NIST increased from 6.0 to 6.2, mWER reduced from 0.42 to 0.41, and, mPER reduced from 0.38 to 0.37. Therefore, the consistent positive evaluation results prove our approach works well in improving speech translation. More than that, we illustrated the change of BLEU for each test utterance in Fig. 6, where we put a mark "x" if the word-lattice-translation's BLEU score was higher than the single-best translation's, a mark "+" if worse, or no sign in the figure if the same BLEU was arrived by both word lattice translation and single-best translation. Each mark corresponds to its utterance number on the x axis. We found there were 43 utterances marked by "x" and 21 by "+", which means that word lattice translation is as twice as likely to get a better translation by our approach.

In the next analysis we highlight part of the test data, for being able to demonstrate the significant contribution of our proposed approaches. The testset of the last section included both single-best correctly-recognized utterances and wrongly-recognized utterances by ASR. In this experiment we only used the single-best wrongly-recognized utterances, composed of 15% of the testset.

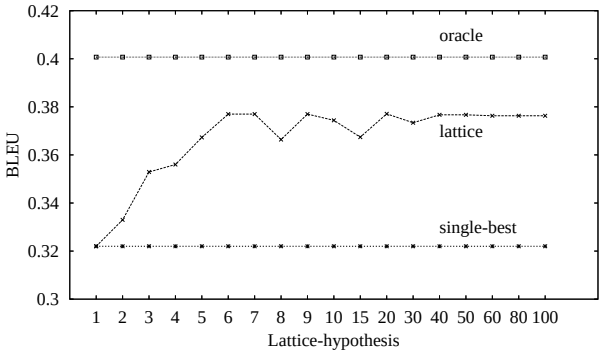

Figure 7: Effect of lattice translation for wrongly recognized utterance

The translation results for this special part are shown in Fig. 7. We used the same experimental settings as the previous experiment and with CM filtering. In the figure, the bottom straight line represents the single-best translation while the top one represents the translations of the ASR oracle results, ASR hypotheses with the best word accuracy. The curve in the middle shows the translation of lattices as the increasing lattice-hypothesis. We found a significant translation improvement by lattice translation. The BLEU score improved from 0.32 when lattice-hypothesis=1 to 0.38 when lattice-hypothesis=20, achieving 75% of the maximum improvement made by the oracle translations, 0.40.

7. Conclusions

This paper described our work on improving speech translation performance. We implemented a new speech translation system, WLT, that can translate speech recogni-

tion word lattice directly. To reduce computation cost and improve translation quality we used a word lattice size reduction method. While all these techniques improved the speech translation, we found the largest gain was achieved by applying the confidence measure in WLT. By choosing ASR hypotheses based on the posterior probability, speech translation qualities were improved to a new higher level. Our work of using the confidence measure is the first attempt in speech translation. We found it is a promising approach for speech translation improvements.

While the main features in the translation module come from IBM Model 4, our translation model is not a pure IBM Model 4 but a log-linear model incorporating the features derived by IBM Model 4. The feature of posterior probability is also integrated in the log-linear model, representing the contributions of the ASR acoustic model and language model.

The WLT developed in this work for speech translation is an extension of a word-based statistical machine translation system, written for text translation. This word-based SMT system achieved comparable performance with the phrase-based translation approach in a recent evaluation [10].

Currently, we only use word-to-word pairs in the log-linear models, though we believe the system will be improved further if we integrate phrase-to-phrase translation pairs. All the proposed approaches, such as word lattice reduction and confidence measure filtering, can be applied to phrase-based translator easily.

8. References

[1] H. Ney, "Speech translation: Coupling of recognition and translation," in *Proc. of ICASSP 1999*, vol. 1, Phoenix, AR, March 1999, pp. 517–520.

[2] Y. Gao, "Coupling vs. unifying: Modeling techniques for speech-to-speech translation," in *Proc. of EuroSpeech 2003*, Geneva, 2003, pp. 365–368.

[3] F. Casacuberta, E. Vidal, and J. M. Vilar, "Architectures for speech-to-speech translation using finite-state models," in *Proc. of speech-to-speech translation workshop*, Philadelphia, PA, July 2002, pp. 39–44.

[4] F. Casacuberta, H.Ney, F.J.Och, E.Vidal, J.M.Vilar, S.Barrachina, I.Garcia-Varea, D.Llorens, C.Martinez, S.Molau, F.Nevada, M.Pastor, D.Pico, A.Sanchis, and C.Tillmann, "Some approaches to statistical and finite-state speech-to-speech translation," in *Computer Speech and Language*, 2004, pp. 25–47.

[5] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[6] S. Vogel, H.Ney, and C.Tillman, "HMM word-based alignment in statistical machine translation," Copenhagen, Denmark, 1996.

[7] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proc. of ACL'2003*, 2003, pp. 160–167.

[8] N. Ueffing, F. J. Och, and H. Ney, "Generation of word graphs in statistical machine translation," in *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP02)*, Philadelphia, PA, July 2002, pp. 156–163.

[9] G. Kikui, E. Sumita, T. Takezawa, and S. Yamamoto, "Creating corpora for speech-to-speech translation," in *Proc.of EUROSPEECH 2003*, Geneva, 2003, pp. 381–384.

[10] Y. Akiba, M. Federico, N. Kando, H. Nakaiwa, M. Paul, and J. Tsujii, "Overview of the IWSLT04 evaluation campaign," in *Proc. of IWSLT04*, ATR, Kyoto, Japan, 2004.