IMPROVED STATISTICAL MODELS FOR SMT-BASED SPEAKING STYLE TRANSFORMATION

Graham Neubig, Yuya Akita, Shinsuke Mori, Tatsuya Kawahara

Kyoto University, School of Informatics
Sakyo-ku, Kyoto, 606-8501, Japan

ABSTRACT

Automatic speech recognition (ASR) results contain not only ASR errors, but also disfluencies and colloquial expressions that must be corrected to create readable transcripts. We take the approach of statistical machine translation (SMT) to "translate" from ASR results into transcript-style text. We introduce two novel modeling techniques in this framework: a context-dependent translation model, which allows for usage of context to accurately model translation probabilities, and log-linear interpolation of conditional and joint probabilities, which allows for frequently observed translation patterns to be given higher priority. The system is implemented using weighted finite state transducers (WFST). On an evaluation using ASR results and manual transcripts of meetings of the Japanese Diet (national congress), the proposed methods showed a significant increase in accuracy over traditional modeling techniques.

Index Terms— speaking style transformation, log-linear models, weighted finite state transducers

1. INTRODUCTION

The task of automatic speech recognition (ASR) is conventionally modeled as finding the verbatim utterance V given the acoustic signal X. A statistical model is created, and a decoder searches for utterance $\hat{V}$ that maximizes $P(V|X)$. However, ASR results often contain disfluencies, redundant or colloquial expressions, and dropped words, in addition to ASR errors. These phenomena must be corrected in order to create readable, natural transcripts from ASR results.

A reasonably large body of work has been conducted on correcting these phenomena automatically, with a particular focus on disfluency deletion and punctuation insertion [1, 2, 3]. However, in addition to disfluency deletion and punctuation insertion, human editors make a number of other edits, including correction of colloquial expressions and dropped words. Handling these phenomena is particularly important in formal settings such as public speeches and congressional meetings, where disfluencies and repairs are less frequent than in conversational speech, but the resulting transcript must be grammatically and stylistically correct.

Previous research on handling the arbitrary transformations necessary for creation of formal transcripts has used techniques from statistical machine translation (SMT), treating verbatim and clean transcripts as different languages and "translating" between them. Shitaoka et al [4] presented a noisy-channel model for speaking style transformation (SST). We expanded this model through a weighted finite state transducer (WFST) implementation and the introduction of a variety of features in a log-linear framework [5].

This paper addresses enhancement of the translation model for finite-state SMT-based SST. First, we review the traditional noisy-channel and a context-dependent joint probability model previously proposed for finite-state machine translation [6]. Then, we propose two improvements to the existing methods. The first method allows for use of a context-dependent translation model in the noisy-channel framework by transforming context-dependent joint probabilities into conditional probabilities. The second method allows greater emphasis to be placed on frequent translation patterns by log-linearly interpolating the joint and conditional probability models. An evaluation is performed on committee meetings from the Japanese Diet (national congress), and the proposed methods show significant improvements in accuracy for both manual transcripts and ASR results.

2. MODELS FOR SPEAKING STYLE TRANSFORMATION (SST)

2.1. Noisy-Channel Modeling

SMT-based SST transforms an actual utterance (or ASR results) V into transcript-style text W by creating a statistical model for $P(W|V)$, and searching for the $\hat{W}$ that maximizes $P(W|V)$ for any given V. A parallel corpus of aligned sentences is used to estimate the parameters of the model. Because the size of available parallel corpora is often dwarfed by the size of available clean transcripts, Bayes' law is used to decompose $P(W|V)$ into the translation model (TM) probability $P_t(V|W)$ and language model (LM) probability $P_l(W)$

$$\hat{W} = \underset{W}{\operatorname{argmax}} P_t(V|W)P_l(W). \quad (1)$$

While the TM must be trained on a parallel corpus (indicated by the subscript t), the LM can utilize a larger body of clean transcripts (subscript l). Models decomposed in this manner are often called *noisy-channel* models, and are used in most previous research on SST [1, 3, 4].

Sentence TM probabilities can be approximated as the product of word TM probabilities to simplify parameter estimation

$$P_t(V|W) \approx \prod_i P_t(v_i|w_i). \quad (2)$$

Word TM probabilities are determined using maximum likelihood estimation.

To handle insertions and deletions, the empty string ϵ is treated as a word in the vocabulary, and probabilities $P_t(v|\epsilon)$ and $P_t(\epsilon|w)$ are calculated. To handle one-to-many substitutions (e.g. "don't" → "do not"), common multi-word phrases are treated as individual vocabulary words. The segmentation of the target sentence into these words is determined by a unigram segmentation model.

2.2. Joint Probability Modeling

While word translation probabilities were assumed to be independent in the previous section, in many cases the translation probability is actually highly context dependent (e.g. whether "like" is a filler or function word, etc.). One method for expressing context directly in the TM is the GIATI method [6]. GIATI skips the step of noisy-channel decomposition and directly models the joint probability $P_t(W, V)$. By limiting the search space so that V is the source sentence,

$$\hat{W} = \underset{W}{\text{argmax}} P_t(W, V)$$

is ensured to give the same result as Equation (1).

GIATI models the joint probability by assuming alignments are monotonic, an assumption that generally holds in the SST task as there is little non-monotonic permutation. The source sentence $V = v_1, \ldots, v_k$ and target sentence $W = w_1, \ldots, w_k$ are represented as a string of symbols $\Gamma = \gamma_1, \ldots, \gamma_k$, where $\gamma_i = \langle v_i, w_i \rangle$. Using these monotonic alignments, a smoothed n-gram model is trained over a corpus of Γ strings. $P_t(W, V)$ is approximated using the following equation:

$$P_t(W, V) = P_t(\Gamma) \approx \prod_{i=1}^{k} P_t(\gamma_i|\gamma_{i-n+1}, \ldots, \gamma_{i-1}). \quad (3)$$

2.3. Context-Dependent Translation Modeling

While joint probability models provide an effective way to handle context, they also leave no room for use of large-scale non-parallel data through the LM probability $P_l(W)$. We propose a technique for approximating a context-dependent TM probability from GIATI probabilities. This allows for the creation of a model that can both consider context when choosing translation probabilities, and use non-parallel data to compensate for sparsity in the parallel corpus.

We first note that $P_t(V|W)$ can be modeled sequentially:

$$\begin{aligned} P_t(V|W) &= \prod_{i=1}^{k} P_t(v_i|v_1, \ldots, v_{i-1}, w_1, \ldots, w_k,) \\ &= \prod_{i=1}^{k} P_t(v_i|\gamma_1, \ldots, \gamma_{i-1}, w_i, \ldots, w_k). \end{aligned}$$

Further, we assume that v_i does not depend on any w greater than w_i, and use an n-order Markov model to limit the length of the considered history:

$$P_t(V|W) \approx \prod_{i=1}^{k} P_t(v_i|\gamma_{i-n+1}, \ldots, \gamma_{i-1}, w_i). \quad (4)$$

Equation (4) can further be transformed into

$$P_t(V|W) \approx \prod_{i=1}^{k} \frac{P_t(\gamma_i|\gamma_{i-n+1}, \ldots, \gamma_{i-1})}{P_t(w_i|\gamma_{i-n+1}, \ldots, \gamma_{i-1})}. \quad (5)$$

The denominator of this equation is the sum of the n-gram probabilities for γ_j where $w_j = w_i$

$$\begin{aligned} P_t(w_i|\gamma_{i-n+1}, \ldots, \gamma_{i-1}) = \\ \sum_{\gamma_j \in \{\tilde{\gamma}:\tilde{w}=w_i\}} P_t(\gamma_j|\gamma_{i-n+1}, \ldots, \gamma_{i-1}). \end{aligned} \quad (6)$$

Because the numerator of Equation (5) and each element in the sum of Equation (6) have the same form as the n-gram probabilities in Equation (3), $P_t(V|W)$ can be estimated using the n-gram probabilities obtained by the GIATI method. This context-dependent model for $P_t(V|W)$ can be used along with LM probability $P_l(W)$ in Equation (1).

2.4. Log-Linear Interpolation with Joint Probabilities

While the conditional model has the advantage of allowing the usage of non-parallel text, it lacks a model of overall translation pattern frequency. For example, if there is a pattern γ_x with counts $c_t(\gamma_x) = 100, c_t(w_x) = 1000$, and a pattern γ_y with counts $c_t(\gamma_y) = 1, c_t(w_y) = 10$, both will be given the same probability

$$P_t(v_x|w_x) = P_t(v_y|w_y) = 0.1$$

even though the less frequent γ_y may simply be the result of semi-random variance in sparse training data. Infrequent patterns are particularly unreliable when dealing with ASR data, which is highly inconsistent.

While $P_t(v_x|w_x)$ and $P_t(v_y|w_y)$ are equal, $P_t(\gamma_x)$ will be 100 times larger than $P_t(\gamma_y)$. Thus, the joint probability can be used as a source of information about translation pattern frequency. We propose a model $M(W, V)$ that uses

Table 1. Size of the test set, and number of transformations necessary for the manual transcriptions and ASR results.

Turns			1,023
Words			300,059
Commas			20,629
Periods			7,196
		Manual	ASR
Deletions	Fillers	22,520	19,468
	Non-fillers	24,450	42,105
Substitutions		4,954	28,503
Insertions		4,584	11,332

log-linear interpolation [7] to combine the LM, TM, and joint probabilities, thus capturing this frequency information

$$M(W,V) = \lambda_1 \log P_l(W) + \lambda_2 \log P_t(V|W) + \lambda_3 \log P_t(W,V) \quad (7)$$

Note that while setting $\lambda_3 = 0$ is an extension to the naive noisy-channel model in Equation (1), setting $\lambda_2 = 0$ and interpolating only the first and third elements is neither theoretically correct nor practical. From the theoretical standpoint, combining $P_l(W)$ and $P_t(W,V)$ can in no way derive $P(W|V)$, the posterior function that we are trying to optimize. Practically, a model created in this way over-aggressively deletes words, resulting in accuracy no better than the standard models. It is for this reason that the conditional model introduced in the previous section is necessary, even when interpolating with the joint probability.

3. EXPERIMENTAL EVALUATION

3.1. Experimental Setup

The proposed system was trained and tested on a corpus of committee meetings of the Japanese Diet (national congress). With the official Diet transcripts as the final target, separate tests were conducted on both manually-created verbatim transcripts and ASR results as input. Punctuation was treated the same as any other word in the translation model, but a symbol indicating pauses of greater than 200ms was included in the ASR output to provide information for punctuation insertion. ASR was performed with a system dedicated to this task [8], and an ASR WER of 17.1% was achieved. A summary of the test set can be found in Table 1.

A corpus of 158M words of official Diet transcripts was used to train the LM. A smaller 2.83M word parallel corpus was used for training of the TM. Log-linear weights were tuned on a set of held-out data consisting of 66.3k words.

3.2. Training/Decoding

The LM was a Kneser-Ney smoothed 3-gram. This configuration was used for all noisy-channel models regardless of the TM order.

Table 2. Each model, whether it is log-linear (LL), and its WER for each TM order. Italics are statistically significantly different from the baseline.

Manual Transcripts (18.62% Unedited)

Model	LL	1-gram	2-gram	3-gram
Noisy		6.51%	*5.33%*	*5.32%*
Noisy LL	✓	*5.99%*	*5.15%*	*5.13%*
Joint		9.89%	*4.70%*	*4.60%*
Noisy+Joint	✓	*5.81%*	*4.12%*	***4.05%***

ASR Results (36.10% Unedited)

Model	LL	1-gram	2-gram	3-gram
Noisy		21.83%	*21.00%*	*21.09%*
Noisy LL	✓	21.63%	*20.97%*	*21.09%*
Joint		28.61%	22.62%	21.98%
Noisy+Joint	✓	*21.32%*	*20.04%*	***20.03%***

The TM n-grams were also Kneser-Ney smoothed, and orders 1-3 were tested (4-grams were inferior to 3-grams for all models). For the system using manual transcripts as input, a parallel corpus of verbatim and official transcripts was used as TM training data. Likewise, when using ASR results as input, a corpus of ASR results and official transcripts was used for training the TM[1]. Word alignment was performed by first aligning words to minimize edit distance, after which words in non-matching sections were aligned using the EM algorithm to minimize the entropy of the joint TM.

The TM, LM, joint, and segmentation models were each expressed as separate WFSTs, and were composed into an overall model using the OpenFst toolkit [9]. Decoding was performed using a beam-search WFST decoder, Kyfd[2]. Log-linear weights were trained using the minimum error rate training tool included in the Moses SMT toolkit [10].

3.3. Effect of Translation Models

Table 2 shows results using four separate models: the noisy-channel model of Equation (5), a noisy-channel model with separate log-linear weights for the TM and LM (Equation (7) with λ_3 fixed at 0), the joint probability model of Equation (3), and the noisy-channel/joint model of Equation (7). Because the 1-gram noisy-channel model is equivalent to traditional noisy-channel models, it is used as a baseline.

The 3-gram noisy-channel/joint model performed best, achieving a WER of 4.05% over manual transcriptions and 20.03% over ASR results. This is a statistically significant gain over the baseline values (6.51% and 21.83% respectively) according to the two-proportion z-test at 99% confidence. The proposed context-dependent TM made a large

[1] A system trained with manual transcripts performed $\approx$ 3% absolute WER worse, largely because models trained on ASR results are better at inserting punctuation, as well as correcting homonyms and ASR errors.

[2] http://www.phontron.com/kyfd

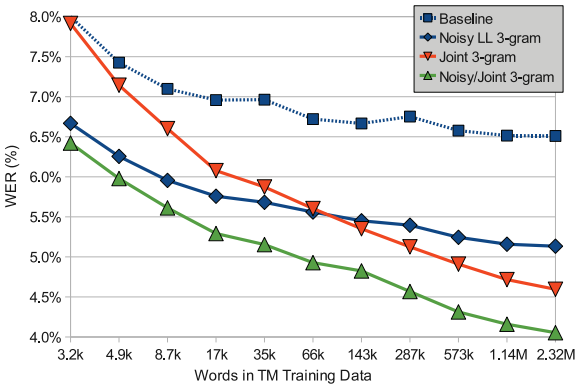

Fig. 1. Effect of corpus size for manual transcripts.

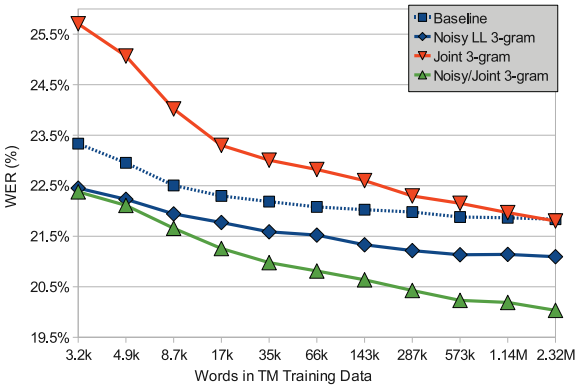

Fig. 2. Effect of corpus size for ASR results.

contribution, resulting in significant gains for both ASR and manual transcripts.

The simple joint probability models (excluding the context-independent 1-gram) performed well on the manual transcripts, but could not outperform the baseline on ASR results. This is because it encountered sparsity issues over the inconsistent data, and was not able to utilize the LM trained with large amounts of data as a fallback. However, when the joint and noisy-channel models were interpolated, further gains were observed, particularly for the 2-gram and 3-gram models where the pattern frequency information provided by the joint probability helped relieve sparseness problems.

3.4. Effect of Corpus Size

We also performed an investigation on the effect of the size of the parallel corpus used in TM training. The results are summarized in Fig. 1 and 2.

First, while noisy-channel models perform better than joint models for small parallel corpora (due to the use of large amounts of clean data), joint models improve at a faster rate as the amount of data increases, even surpassing the noisy-channel model in the manual transcipts of Figure 1. The model that interpolates joint and noisy-channel probabilities displays both good performance on small data and continuing improvment, achieving a low WER over all data sizes.

Second, accuracy continues to improve for the joint models even at 2.32M words, indicating that it may be useful to collect more data. We plan to create more ASR results for training the SST system with ASR input.

Finally, for most models, the error rate decreases rapidly for corpus sizes under 17k words, and more slowly after 17k words. This suggests that common patterns like filler deletions are learned after 17k words, and any additional data after that helps train more difficult context-dependent patterns.

4. CONCLUSION

This paper presented techniques to model context and translation pattern frequency for SMT-based SST. A system using both of these techniques showed a statistically significant improvement over a traditional noisy-channel model for both manual transcripts and ASR results. A promising future research direction is the integration of the SST module with a WFST-based recognition engine to find globally optimal output given acoustic features. We also plan to combine the features introduced in [5] with the translation models presented here and investigate their mutual effect.

5. REFERENCES

[1] M. Honal and T. Schultz, "Correction of disfluencies in spontaneous speech using a noisy-channel approach," in *Proc. EuroSpeech2003*, 2003, pp. 2781–2784.

[2] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper, "Enriching speech recognition with automatic detection of sentence boundaries and disfluencies," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1526–1540, 2006.

[3] S. Maskey, B. Zhou, and Y. Gao, "A phrase-level machine translation approach for disfluency detection using weighted finite state transducers," in *Proc. InterSpeech2006*, 2006, pp. 749–752.

[4] K. Shitaoka, H. Nanjo, and T. Kawahara, "Automatic transformation of lecture transcription into document style using statistical framework," in *Proc. InterSpeech2004*, 2004, pp. 2169–2172.

[5] G. Neubig, S. Mori, and T. Kawahara, "A WFST-based log-linear framework for speaking-style transformation," in *Proc. InterSpeech2009*, 2009, pp. 1495–1498.

[6] F. Casacuberta and E. Vidal, "Machine translation with inferred stochastic finite-state transducers," *Computational Linguistics*, vol. 30, no. 2, pp. 205–225, 2004.

[7] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proc. ACL02*, 2002, pp. 295–302.

[8] Y. Akita, M. Mimura, and T. Kawahara, "Automatic transcription system for meetings of the Japanese national congress," in *Proc. InterSpeech2009*, 2009, pp. 84–87.

[9] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "OpenFst: a general and efficient weighted finite-state transducer library," in *Proc. CIAA '07*, 2007, pp. 11–23.

[10] P. Koehn et al., "Moses: Open source toolkit for statistical machine translation," in *Proc. ACL07*, 2007, pp. 177–180.