Mitigation of data sparsity in classifier-based translation

Emil Ettelaie, Panayiotis G. Georgiou, Shrikanth S. Narayanan
Signal Analysis and Interpretation Laboratory
Ming Hsieh Department of Electrical Engineering
Viterbi School of Engineering
University of Southern California
ettelaie@usc.edu

Abstract

The concept classifier has been used as a translation unit in speech-to-speech translation systems. However, the sparsity of the training data is the bottle neck of its effectiveness. Here, a new method based on using a statistical machine translation system has been introduced to mitigate the effects of data sparsity for training classifiers. Also, the effects of the background model which is necessary to compensate the above problem, is investigated. Experimental evaluation in the context of cross-lingual doctor-patient interaction application show the superiority of the proposed method.

1 Introduction

Statistical machine translation (SMT) methods are well established in speech-to-speech translation systems as the main translation technique (Narayanan et al., 2003; Hsiao et al., 2006). Due to their flexibility these methods provide a good coverage of the dialog domain. The fluency of the translation, however, is not guaranteed. Disfluencies of spoken utterances plus the speech recognizer errors degrade the translation quality even more. All these ultimately affect the quality of the synthesized speech output in the target language, and the effectiveness of the concept transfer.

It is quite common, though, to use other means of translation in parallel to the SMT methods (Gao et al., 2006; Stallard et al., 2006). Concept classification, as an alternative translation method, has been successfully integrated in speech-to-speech translators (Narayanan et al., 2003; Ehsani et al., 2006). A well defined dialog domain, e.g. doctor-patient dialog, can be partly covered by a number of concept classes. Upon a successful classification of the input utterance, the translation task reduces to synthesizing a previously created translation of the concept, as a mere look up. Since the main goal in such applications is an accurate exchange of concepts, this method would serve the purpose as long as the input utterance falls within the coverage of the classifier. This process can be viewed as a quantization of a continuous "semantic" sub-space. The classifier is adequate when the quantization error is small (i.e. the derived concept and input utterance are good matches), and when the utterance falls in the same sub-space (domain) as the quantizer attempts to cover. Since it is not feasible to accurately cover the whole dialog domain (since a large number of quantization levels needed) the classifier should be accompanied by a translation system with a much wider range such as an SMT engine. A rejection mechanism can help identify the cases that the input utterance falls outside the classifier coverage (Ettelaie et al., 2006).

In spite of this short coming, the classifier-based translator is an attractive option for speech-to-speech applications because of its tolerance to "noisy" input and the fluency of its output, when it operates close to its design parameters. In practice this is attainable for structured dialog interactions with high levels of predictability. In addition, it can provide the users with both an accurate feedback and different translation options to choose from. The latter feature, specially, is useful for applications like doctor-patient dialog.

Building a concept classifier starts with identifying the desired concepts and representing them with canonical utterances that express these concepts. A good set of concepts should consist of the ones that are more frequent in a typical interaction in the domain. For instance in a doctor-patient dialog, the utterance "Where does it hurt?" is quite common and therefore its concept is a good choice. Phrase books, websites, and experts' judgment are some of the resources that can be used for concept selection. Other frequently used concepts include those that correspond to basic communicative and social aspects of the interaction such as greeting, acknowledgment and confirmation.

After forming the concept space, for each class,

Coling 2008: Proceedings of the workshop on Speech Processing for Safety Critical Translation and Pervasive Applications, pages 1–4
Manchester, August 2008

utterances that convey its concept must be gathered. Hence, this training corpus would consist of a group of paraphrases for each class. This form of data are often very difficult to collect as the number of classes grow. Therefore, the available training data are usually sparse and cannot produce a classification accuracy to the degree possible. Since the classifier range is limited, high accuracy within that range is quite crucial for its effectiveness. One of the main issues is dealing with data sparsity. Other techniques have also been proposed to improve the classification rates. For example in (Ettelaie et al., 2006) the accuracy has been improved by introducing a dialog model. Also, a background model has been used to improve the discrimination ability of a given concept class model.

In this work a novel method for handling the sparsity is introduced. This method utilizes an SMT engine to map a single utterance to a group of them. Furthermore, the effect of the background model on classification accuracy is investigated.

Section 2 reviews the concept classification process and the background model. In Section 3 the sparsity handling method using an SMT is introduced. Data and experiments are described in Section 4. The results are discussed in Section 5.

2 Concept classifier and background model

The concept classifier based on the maximum likelihood criterion can be implemented as a language model (LM) scoring process. For each class a language model is built using data expressing the class concept. The classifier scores the input utterance using the class LM's and selects the class with highest score. In another word if C is the set of concept classes and $\mathbf{e}$ is the input utterance, the classification process is,

$$\hat{c} = \arg\max_{c \in C} \left\{ P_c\left(\mathbf{e} \mid c\right) \right\} \tag{1}$$

where $P_c(\mathbf{e} \mid c)$ is the score of $\mathbf{e}$ from the LM of class c. The translation job is concluded by playing out a previously constructed prompt that expresses the concept $\hat{c}$ in the target language.

It is clear that a class with limited training data items will have an undertrained associated LM with poor coverage. In practice such a model fails to produce a usable LM score and leads to a poor classification accuracy. Interpolating the LM with a background language model results in a smoother model (Stolcke, 2002) and increases the overall accuracy of the classifier.

The background model should be built from a larger corpus that fairly covers the domain vocabulary. The interpolation level can be optimized for the best performance based on heldout set.

3 Handling sparsity by statistical machine translation

The goal is to employ techniques that limit the effects of data sparsity. What is proposed here is to generate multiple utterances – possibly with lower quality – from a single original one. One approach is to use an SMT to generate n-best lists of translation candidates for the original utterances. Such lists are ranked based on a combination of scores from different models (Ney et al., 2000). The hypothesis here is that for an SMT trained on a large corpus, the quality of the candidates would not degrade rapidly as one moves down the n-best list. Therefore a list with an appropriate length would consist of translations with acceptable quality without containing a lot of poor candidates. This process would result in more data, available for training, at the cost of using noisier data.

Although the source language of the SMT must be the same as the classifier's, its target language can be selected deliberately. It is clear that a language with large available resources (in the form of parallel corpora with the source language) must be selected. For simplicity this language is called the "intermediate language" here.

A classifier in the intermediate language can be built by first generating an n-best list for every source utterance in the classifier's training corpus. Then the n-best lists associated with each class are combined to form a new training set. The class LM's are now built from these training sets rather than the original sets of the source utterances.

To classify a source utterance $\mathbf{e}$, first the SMT is deployed to generate an n-best list (in the intermediate language) from it. The list will consist of candidates $\mathbf{f}_1$, $\mathbf{f}_2$,..., $\mathbf{f}_n$. The classification process can be reformulated as,

$$\hat{c} = \arg\max_{c \in C} \left\{ \prod_{i=1}^{n} \tilde{P}_c\left(\mathbf{f}_i \mid c\right) \right\} \tag{2}$$

Here, $\tilde{P}_c(\mathbf{f}_i \mid c)$ is the score of the i^{th} candidate $\mathbf{f}_i$ from the LM of class c. The scores are considered in the probability domain.

The new class LM's can also be smoothed by interpolation with a background model in the intermediate language.

4 Data and Experiments

4.1 Data

The data used in this work were originally collected for, and used in, the Transonics project (Narayanan et al., 2003) to develop an English/Farsi speech-to-speech translator in the doctor-patient interaction domain. For the doctor side, 1,269 concept classes were carefully chosen using experts' judgment and medical phrase books. Then, for each concept, English data were collected from a website, a web-based game, and multiple paraphrasing sessions at the Information Sciences Institute of the University

	Conventional (baseline)	n-best length			
		100	500	1,000	2,000
Accuracy [%]	74.9	77.4	77.5	76.8	76.4
Relative error reduction [%]	0.0	10.0	10.4	7.6	6.0
Accuracy in 4-best [%]	88.6	90.7	91.0	91.3	90.5
Relative error reduction [%]	0.0	18.4	21.1	23.7	16.7

Table 1: Classification accuracy for the conventional method and the proposed method with different lengths of n-best list

of Southern California. The total size of the data set consists of 9,893 English phrases.

As the test corpus for this work, 1,000 phrases were randomly drawn from the above set and the rest were used for training. To make sure that the training set covered every class, one phrase per class was excluded from the test set selection process.

To generate the n-best lists, a phrase based SMT (Koehn et al., 2003) was used. The intermediate language was Farsi and the SMT was trained on a parallel English/Farsi corpus with 148K lines (1.2M words) on the English side. This corpus was also used to build the classification background models in both languages. The SMT was optimized using a parallel development set with 915 lines (7.3K words) on the English side.

4.2 Classification Accuracy Measures

Classifier accuracy is often used as the the quality indicator of the classification task. However, it is common in the speech-to-speech translation systems to provide the user with a short list of potential translations to choose from. For example the user of system in (Narayanan et al., 2003) is provided with the top four classifier outputs. In such cases, it is practically useful to measure the accuracy of the classifier within its n-best outputs (e.g., $n = 4$ for the above system). In this work the classification accuracy was measured on both the single output and the 4-best outputs.

4.3 Experiments

To compare the proposed method with the conventional classification, a classifier based on each method was put to test. In the proposed method, it is expected that the accuracy is affected by the length of the n-best lists. To observe that, n-best lists of lengths 100, 500, 1000, and 2000 were used in the experiments. The results are shown in Table 1. In all of the above experiments the background interpolation factor was set to 0.9 which is close to the optimum value obtained in (Ettelaie et al., 2006).

To examine the effect of the background model, the conventional and proposed methods were tried with different values of the interpolation factor λ (the background model is weighted by $1 - \lambda$). For the conventional method the length of the n-best list was set to 500. Figure 1 shows the accuracy changes with respect to the interpolation factor for these two methods.

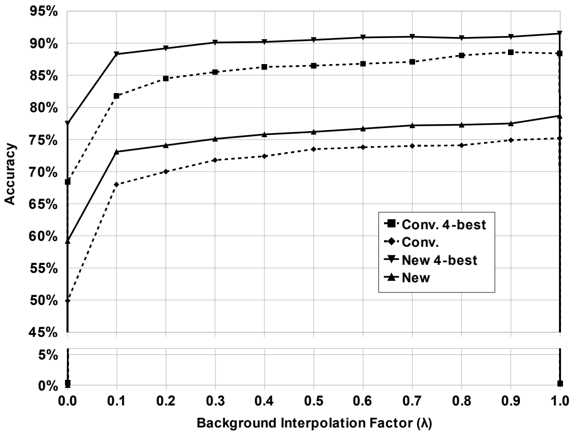

Figure 1: The effect of background model on classification accuracy

5 Discussion

Table 1 shows the advantage of the proposed method over the conventional classification with a relative error rate reduction up to 10.4% (achieved when the length of the SMT n-best list was 500). However, as expected, this number decreases with longer SMT n-best lists due to the increased noise present in lower ranked outputs of the SMT.

Table 1 also shows the accuracy within 4-best classifier outputs for each method. In that case the proposed method showed an error rate which was relatively 23.7% lower than the error rate of the conventional method. That was achieved at the peak of the accuracy within 4-best, when the length of the SMT n-best list was 1,000. In this case too, further increase in the length of the n-best list led to an accuracy degradation as the classifier models became noisier.

The effect of the background model on classifier accuracy is shown in Figure 1. The figure shows the one-best accuracy and the accuracy within 4-best outputs, versus the background interpolation factor (λ) for both conventional and proposed methods. As the curves indicate, with λ equal to zero the classifier has no discriminating feature since all the class scores are driven solely from the background model. However, a slight increase in λ, leads to a large jump in the accuracy. The reason is that the background model was built from a large general domain corpus and hence, had no bias toward any of the classes. With a small λ, the score from the background model dominates the overall class scores. In spite of that, the score differences caused by the class LM's are notable in improving the classifier performance.

As λ increases the role of the class LM's becomes more prominent. This makes the classifier models more discriminative and increases its accuracy as shown in Figure 1. When the factor is in the close vicinity of one, the smoothing effect of the background model diminishes and leaves the classes with spiky models with very low vocabulary coverage (lots of zeros). This leads to a rapid drop in accuracy as λ reaches one.

Both the conventional and proposed methods follow the above trend as Figure 1 shows, although, the proposed method maintains its superiority throughout the range of λ that was examined. The maximum measured accuracies for conventional and proposed methods were 75.2% and 78.7% respectively and was measured at $\lambda = 0.999$ for both methods. Therefore, the error rate of the proposed method was relatively 14.1% lower than its counterpart from the conventional method.

Figure 1 also indicates that when the accuracy is measured within the 4-best outputs, again the proposed method outperforms the conventional one. The maximum 4-best accuracy for the conventional method was measured at the sample point $\lambda = 0.9$ and was equal to 88.6%. For the proposed method, that number was measured as 91.5% achieved at the sample point $\lambda = 0.999$. In another words, considering the 4-best classifier outputs, the error rate of the proposed method was relatively 25.4% lower.

6 Conclusion

The proposed language model based method can be used to improve the accuracy of the concept classifiers specially in the case of sparse training data. It outperformed the conventional classifier, trained on the original source language paraphrases, in the experiments. With this method, when the input utterance is within the classification domain, the classifier can be viewed as a filter that produces fluent translations (removes the "noise") from the SMT output.

The experiments also emphasized the importance of the background model, although indicated that the classification accuracy was not very sensitive to the value of the background interpolation factor. This relieves the developers from the fine tuning of that factor and eliminates the need for a development data set when a suboptimal solution is acceptable.

We believe that significant improvements to the technique can be made through the use of weighted n-best lists based on the SMT scores. In addition we believe that using a much richer SMT engine could provide significant gains through increased diversity in the output vocabulary. We intend to extend on this work through the use of enriched, multilingual SMT engines, and the creation of multiple classifiers (in several intermediate languages).

7 Acknowledgment

This work was supported in part by funds from DARPA.

References

Ehsani, F., J. Kinzey, D. Master, K. Sudre, D. Domingo, and H. Park. 2006. S-MINDS 2-way speech-to-speech translation system. In *Proc. of the Medical Speech Translation Workshop, Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT)*, pages 44–45, New York, NY, USA, June.

Ettelaie, E., P. G. Georgiou, and S. Narayanan. 2006. Cross-lingual dialog model for speech to speech translation. In *Proc. of the Ninth International Conference on Spoken Language Processing (ICLSP)*, pages 1173–1176, Pittsburgh, PA, USA, September.

Gao, Y., L. Gu, B. Zhou, R. Sarikaya, M. Afify, H. Kuo, W. Zhu, Y. Deng, C. Prosser, W. Zhang, and L. Besacier. 2006. IBM MASTOR SYSTEM: Multilingual automatic speech-to-speech translator. In *Proc. of the Medical Speech Translation Workshop, Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT)*, pages 53–56, New York, NY, USA, June.

Hsiao, R., A. Venugopal, T. Kohler, Y. Zhang, P. Charoenpornsawat, A. Zollmann, S. Vogel, A. W. Black, T. Schultz, and A. Waibel. 2006. Optimizing components for handheld two-way speech translation for an English-Iraqi Arabic system. In *Proc. of the Ninth International Conference on Spoken Language Processing (ICLSP)*, pages 765–768, Pittsburgh, PA, USA, September.

Koehn, P., F. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT)*, volume 1, pages 48–54, Edmonton, AB, Canada, May-June.

Narayanan, S., S. Ananthakrishnan, R. Belvin, E. Ettelaie, S. Ganjavi, P. Georgiou, C. Hein, S. Kadambe, K. Knight, D. Marcu, H. Neely, N. Srinivasamurthy, D. Traum, and D. Wang. 2003. Transonics: A speech to speech system for English-Persian interactions. In *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 670–675, St.Thomas, U.S. Virgin Islands, November-Decmeber.

Ney, H., S. Nießen, F. J. Och, C. Tillmann, H. Sawaf, and S. Vogel. 2000. Algorithms for statistical translation of spoken language. *IEEE Trans. on Speech and Audio Processing, Special Issue on Language Modeling and Dialogue Systems*, 8(1):24–36, January.

Stallard, D., F. Choi, K. Krstovski, P. Natarajan, R. Prasad, and S. Saleem. 2006. A hybrid phrase-based/statistical speech translation system. In *Proc. of the Ninth International Conference on Spoken Language Processing (ICLSP)*, pages 757–760, Pittsburgh, PA, USA, September.

Stolcke, A. 2002. SRILM - an extensible language modeling toolkit. In *Proc. of the International Conference on Spoken Language Processing (ICSLP)*, pages 901–904, Denver, CO, USA, September.