SUMMARY  This paper presents a technique for class-dependent decoding for statistical machine translation (SMT). The approach differs from previous methods of class-dependent translation in that the class-dependent forms of all models are integrated directly into the decoding process. We employ probabilistic mixture weights between models that can change dynamically on a sentence-by-sentence basis depending on the characteristics of the source sentence. The effectiveness of this approach is demonstrated by evaluating its performance on travel conversation data. We used this approach to tackle the translation of questions and declarative sentences using class-dependent models. To achieve this, our system integrated two sets of models specifically built to deal with sentences that fall into one of two classes of dialog sentence: questions and declarations, with a third set of models built with all of the data to handle the general case. The technique was thoroughly evaluated on data from 16 language pairs using 6 machine translation evaluation metrics. We found the results were corpus-dependent, but in most cases our system was able to improve translation performance, and for some languages the improvements were substantial.

1. Introduction

Topic-dependent modeling has proven to be an effective way to improve the quality of models in speech recognition (Iyer and Osendorf, [1]; Carter, [2]). Recently, experiments in the field of machine translation (Hasan and Ney, [3]; Yamamoto and Sumita, [4]; Finch et al. [5], Foster and Kuhn, [6]) have shown that class-specific models are also useful for translation.

In the method proposed by Yamamoto and Sumita [4], topic dependency was implemented by partitioning the data into sets before the decoding process started and subsequently decoding these sets independently using different models that were specific to the class predicted for the source sentence by a classifier that was run over the source sentences in a pre-processing pass. Our approach is in many ways a generalization of this work. Our technique allows the use of multiple-model sets within the decoding process itself. The contributions of each model set can be controlled dynamically during the decoding through a set of interpolation weights. These weights can be changed on a sentence-by-sentence basis. The previous approach is, in essence, the case where the interpolation weights are either 1 (indicating that the source sentence is the same topic as the model) or 0 (the source sentence is a different topic).

One advantage of our proposed technique is that it is a soft approach. That is, the source sentence can belong to multiple classes to varying degrees. In this respect our approach is similar to that of Foster and Kuhn [6], but we used a classifier to determine a vector of probabilities representing class-membership, rather than distance-based weights. These probabilities were used directly as the mixture weights for the respective models in an interpolated model-set. A second difference between our approach and that of Foster and Kuhn, is that we include models built from all of the data along with the set of class-specific models. We refer to this model as the General SMT system in the rest of this paper.

Our approach differs from all previous approaches in the models that are class-dependent. Hasan and Ney [3] used only a class-dependent language model. Both Yamamoto and Sumita [4] and Foster and Kuhn [6], extended this to include the translation model. In our approach we combine all of the models in the SMT system, including the distortion and target-length models, within a single framework.

The contribution of this paper is two-fold. The first is the proposal of a technique for combining multiple SMT systems in a weighted manner to allow probabilistic soft weighting between topic-dependent models for all models in the system. The second is the application of this technique to improve the quality of dialog systems by building and combing class-based models for interrogative and declarative sentences.

For the purposes of this paper, we make the distinction between interrogative sentences and those which are not. For the sake of simplicity, in this article interrogative sentences are called questions, while all others are called declarations. The two-letter abbreviations used for the languages mentioned in this paper are given in the key that is Table 1.

Table 1  Key to the abbreviations of the languages used in our experiments.

| Language | Abbreviation |
|----------|--------------|
| Arabic   | ar           |
| Danish   | da           |
| German   | de           |
| Spanish  | es           |
| French   | fr           |
| Indonesian | id       |
| Italian  | it           |
| Japanese | ja           |
| Korean   | ko           |
| Malaysian| ms           |
| Dutch    | nl           |
| Portuguese| pt        |
| Russian  | ru           |
| Thai     | th           |
| Vietnamese| vi         |
| Chinese  | zh           |
2. Background

In this section we present a brief overview of the statistical machine translation technology used to build the systems described in this paper.

Typically an SMT system (for example those based on the PHARAOH [7] and MOSES [8] decoders) is built from two corpora: a small bilingual corpus of sentence-aligned sentence pairs, and a large monolingual corpus of data from the target language. The reason for this discrepancy in corpus size is simply that bilingual data is expensive to produce.

The SMT systems used in our experiments are composed of two basic models: the translation model and the language model. The translation model, models the process of generating the target sentence from the source sentence, and the language model ensures that the target sentence is a likely sequence of words in the target language. The translation model is trained using the bilingual corpus, and the language model is trained using the monolingual corpus. The translation model can be further decomposed into sub-models that model word/phrase translations, target sentence length, word/phrase reordering and so on. The models are combined in a log-linear manner, each model being weighted exponentially by weights that are tuned to minimize an error function, on held-out data (see [9]).

During the training of the translation model, the words in the sentences are aligned automatically using the EM algorithm and bilingual word sequence pairs are extracted from the corpus by applying heuristics with reference to the word alignments. These bilingual phrase pairs are then assigned phrase-translation probabilities according to their relative frequency, and are stored in a structure called the phrase table. During translation, the source language sides of the bilingual phrase pairs from the phrase table are used to translate subsequences of the input word sequence, and as part of the same process, the target side of the phrase pairs generates the translation. The source sentence can be translated in any order, but the target word sequence is always generated from left to right. This non-monotone translation of the source sentence allows for phrase reordering during translation.

3. System Description

3.1 Overview

Our proposed method is a weighted combination of a number of SMT systems: a general SMT system that has been trained on all of the available data, and a set of topic-dependent systems that have been trained on subsets of the training data that correspond to the respective topics for these models. In our experiments, we used two topic-dependent systems: one trained on questions, the other trained on declarations.

3.2 Question Prediction

3.2.1 Outline of the Problem

Given a source sentence of a particular type (questions or declarations in our case), we wish to ensure that the target sentence generated is of an appropriate class. Note that this does not necessarily mean that given a question in the source, a question should be generated in the target. However, it seems reasonable to assume that, intuitively at least, one should be able to generate a target question from a source question, and a target declaration from a source declaration. This is reasonable because the role of a machine translation engine is not to be able to generate every possible translation from the source, but to be able to generate one acceptable translation. This assumption leads us to two plausible ways to proceed.

1. To predict the class of the source sentence, and use this to constrain the decoding process used to generate the target
2. To predict the class of the target

In our experiments, we chose the second method, as it seemed the most correct, but feel there is some merit in both strategies.

3.2.2 The Maximum Entropy Classifier

We used a Maximum Entropy (ME) classifier to determine which class to which the input source sentence belongs using a set of lexical features. That is, we use the classifier to set the mixture weights of the class-specific models. In recent years such classifiers have produced powerful models utilizing large numbers of lexical features in a variety of natural language processing tasks, for example [10], [11]. An ME model is an exponential model with the following form:

\[ p(t,c) = \gamma \prod_{k=0}^{K} \alpha_k f_k(c,t) p_0 \]

where:

- \( t \) is class being predicted;
- \( c \) is the context of \( t \);
- \( \gamma \) is a normalization coefficient;
- \( K \) is the number of features in the model;
- \( \alpha_k \) is the weight of feature \( f_k \);
- \( f_k \) are binary feature functions;
- \( p_0 \) is the default model.

We used the set of all n-grams (n \( \leq 3 \)) occurring in the source sentences as features to predict the sentences class. Additionally we introduced beginning of sentence tokens (\(<s>\)) and end of sentence tokens (\(</s>\)) into the word sequence to distinguish n-grams occurring at the start and end of sentences from those occurring within the sentence.
This was based on the observation that “question words” or words that indicate that the sentence is a question will frequently be found either at the start of the sentence (as in the wh- <what, where, when> words in English or the -kah words in Malay <apakah, dimanakah, kapankah>), or at the end of the sentence (for example the Japanese ka or the Chinese ma. In fact, in earlier models we used features consisting of n-grams occurring only at the start and end of the source sentence. These classifiers performed quite well (approximately 4% lower than the classifiers that used features from all of the n-grams in the source), but an error analysis showed that n-grams from the interior of the sentence were necessary to handle sentences such as: “excuse me please where is ...”. A simple example sentence and the set of features generated from the sentence is shown in Fig. 1.

We used the ME modeling toolkit of [12] to implement our ME models. The models were trained by using L-BFGS parameter estimation, and a Gaussian prior was used for smoothing during training.

3.2.3 Forcing the Target to Conform

Before adopting the mixture-based approach set out in this paper, we first pursued an obvious and intuitively appealing way of using this classifier. We applied it as a filter to the output of the decoder, to force source sentences that the classifier predicts should generate questions in the target to actually generate questions in the target. This approach was unsuccessful due to a number of issues. We took the n-best output from the decoder and selected the highest translation hypothesis on the list that had agreement on class according to source and target classifiers. The issues we encountered included, too much similarity in the n-best hypotheses, errors of the MT system were correlated with errors of the classifier, and the number of cases that were corrected by the system was small < 2%. As a consequence, the method proposed in this paper was preferred.

3.3 System Architecture

3.3.1 The Training Process

Figure 2 shows the way the corpora were used to train our system. The sentences in the corpus were labeled according to whether or not the source sentence translates to a question in the target language. We used punctuation (a sentence-final ? character) on the target-side as the ground truth as to the class of the target sentence. Neither punctuation nor case information was used for any other purpose in the experiments.

The bilingual corpus was partitioned into two classes: Questions and Declarations. Then the whole corpus and each class were further sub-divided into training and development sets. In each case, 1000 sentences were set aside as development data, and the remainder was used for training. This data was then used to build three SMT systems (shown in Fig. 3): one for each class (labeled *Question specific SMT system* and *Declaration specific SMT system*), and one using the data from both classes (labeled *General SMT system*). A probabilistic classifier (described in the next section) was also trained from the full set of training data.

3.3.2 Decoding with Interpolation

The process of decoding for the model interpolating machine translation decoder is shown in Fig. 3. This decoder has the capability to linearly interpolate all of the models from all of the sub-systems according to a vector of interpo-
Fig. 3 The architecture of the class-based SMT system used in our experiments.

Fig. 4 Example sentences from the English BTEC1 corpus.

Excuse me, what time is the next train to Glasgow?
It takes about 30 minutes.
I feel great now, can you send up some champagne?

4. Experiments

4.1 Experimental Data

To evaluate the proposed technique, we conducted experiments on a travel conversation corpus. The experimental corpus was the first part of the BTEC corpus (BTEC1) [13] and used English as the target and each of the other languages as source languages. This corpus consists of short sentences that are the kinds of sentences one would expect to find in a phrasebook for travelers. Example sentences are shown in Fig. 4.

The training, development, and evaluation corpus statistics are shown in Table 2. Here, the number of sentences is the same as these values for all source languages. The number of words in the source language differs, and depends on the segmentation granularity.

The evaluation corpus had sixteen reference translations per sentence. The training corpus is similar in character to the IWSLT06 Evaluation Campaign on Spoken Language Translation [14] J-E open track, and the evaluation corpus was used as the IWSLT05 evaluation set.

4.2 Experimental Conditions

4.2.1 Decoding Conditions

The decoder used in the experiments, CleopATRa is an in-house phrase-based statistical decoder that can operate on the same principles as the PHARAOH [7] and MOSES [8] decoders. The decoder was configured to produce near-identical output to MOSES for these experiments. The decoder was modified in order to handle multiple-sets of models, accept weighted input, and to incorporate the dynamic interpolation process into the decoding.

For tuning of the decoder’s parameters, minimum error training [9] with respect to the BLEU score was conducted using the respective development corpus. A 5-gram language model, built using the SRI language modeling toolkit [15] with Witten-Bell smoothing was used. The model included a length model, and also the simple
Table 2  The corpus statistics of the target language corpus (en).

|                  | Questions + Decls. | Questions | Declarations | Test |
|------------------|--------------------|-----------|--------------|------|
|                  | Train   | Dev  | Train   | Dev  | Train    | Dev  |
| Sentences        | 161317  | 1000 | 69684  | 1000 | 90633    | 1000 | 510    |
| Words            | 1001671 | 6112 | 445676 | 6547 | 549375   | 6185 | 3169   |

Table 3  The classification accuracy (%) of the classifier used to predict whether or not an input sentence either is or should give rise to a question in the target.

| Source Language | English Punctuation | Own Punctuation |
|-----------------|---------------------|-----------------|
| ar              | 98.0                | N/A             |
| da              | 97.3                | 98.0            |
| de              | 98.1                | 98.6            |
| en              | 98.9                | 98.9            |
| es              | 96.3                | 96.7            |
| fr              | 97.7                | 98.7            |
| id              | 97.9                | 98.5            |
| it              | 94.9                | 95.4            |
| ja              | 94.1                | N/A             |
| ko              | 94.2                | 99.4            |
| ms              | 98.1                | 99.0            |
| nl              | 98.1                | 99.0            |
| pt              | 96.2                | 96.0            |
| ru              | 95.9                | 96.6            |
| th              | 98.2                | N/A             |
| vi              | 97.7                | 98.0            |
| zh              | 93.2                | 98.8            |

Fig. 5  Graph showing the BLEU score on the development set plotted against the General SMT system’s interpolation weight.

distance-based distortion model used by the PHARAOH decoder [7].

Perhaps the largest concerns about the proposed approach come from the heavy resource requirements that could potentially occur when dealing with large numbers of models. However, one important characteristic of the decoder used in our experiments is its ability to leave its models on disk, loading only the parts of the models necessary to decode the sentence in hand. This reduced the memory overhead considerably when loading multiple models, without noticeably affecting decoding time. Moreover, it is also possible to pre-compute the interpolated probabilities for most of the models for each sentence before the search commences, reducing both search memory and processing time.

4.2.2 Tuning the Interpolation Weights

The interpolation weight for the General SMT system was tuned by maximizing the BLEU score on the development set by grid search over a set of weights ranging from 0 to 1 in increments of 0.1. Figure 5 shows the behavior of two of our models with respect to their weight parameter. A weight of here 0 means there is no contribution from the general system, a weight of 1 means there is no contribution from the class-dependent systems.

4.3 Evaluation

To obtain a balanced view of the merits of our proposed approach, in our experiments we used 6 evaluation techniques to evaluate our systems. These were: BLEU [16], NIST [17], WER (Word Error Rate), PER (Position-independent WER), GTM (General Text Matcher), and METEOR [18]. The proposed system was evaluated with respect to a Baseline system consisting only of the General SMT System shown in Fig. 3. The Baseline system is in fact a special case of the proposed system in which the interpolation weight for the General SMT system is 1.

4.3.1 Classification Accuracy

The performance of the classifier (from 10-fold cross-validation on the training set) is shown in Table 3. Ten separate experiments where run, in each 90% of the training data was used to train the classifier and 10% was held out for evaluation. In each experiment, there was no overlap of the test set data with the test set data of the other
experiments. The results in Table 3 are the average over all 10 experiments. We give classification accuracy figures for predicting both source (same language) and target (English) punctuation. Here “same language” means predicting the sentence final punctuation of sentences (from which punctuation has been removed) in a language. When predicting the target’s punctuation, the same source sentence with punctuation removed is used as context for the classifier, however in this case it is the sentence final punctuation of the target sentence that is predicted.

4.3.2 Machine Translation Quality

The performance of the SMT systems trained and tested on the corpora shown in Table 2 is given in Table 5. Example machine translation outputs from the baseline and proposed systems are shown in Table 4. It is clear from Table 5 that for most of the experimental conditions the proposed system outperformed the Baseline system described in Sect. 4.3 that was trained on all of the data. For those metrics in which performance degraded, in all-but-one the results were statistically insignificant, and in all cases most of the other MT evaluation metrics showed an improvement.

4.3.3 Comparison to Previous Methods

We ran an experiment to compare our proposed method to an instance of our system that used hard weights, again on the corpora shown in Table 2. The aim was to come as close as possible within our framework to the system proposed by Yamamoto and Sumita [4]. We used weights of 1 and 0, instead of the classification probabilities to weight the class-specific models. To achieve this, we thresholded the probabilities from the classifier such that probabilities < 0.5 gave a weight of 1, otherwise a weight of 0 was used. The performance of this system is shown in Table 6 under the column heading Hard. In all-but-one of the conditions this system was outperformed by or equal to the proposed approach.

The column labeled “No Classifier” in Table 6 illustrates the effectiveness of the classifier in our system. These results show the effect of using equal weights (0.5) to interpolate between the Question and Declaration models. This system, although not as effective as the system with the classifier, gave a respectable performance.

5. Discussion

5.1 The Classifier

Unsurprisingly, all systems were better at predicting their own punctuation as can be seen in Table 3. The poorer scores in the table might reflect linguistic characteristics (perhaps questions in the source language are often expressed as statements in the target), or characteristics of the corpus itself. For all languages the accuracy of the classifier seemed satisfactory, especially considering the possibility of inconsistencies in the corpus itself (and therefore our test data for this experiment).

5.2 The Effects of Model Interpolation

It is important to bear in mind that the Baseline system is in fact a special case of the Proposed system in which the weight of the General SMT system is 1. In this case only the General SMT system makes a contribution to the decoder’s score, the weights for the Question- and Declaration-specific systems being 0. Those systems (ar, es and zh) that our technique was not able to improve all learned weights of 1 for the General SMT system, and this explains why these systems had identical evaluation scores to the Baseline system.

Some of the machine translation evaluation schemes show inconsistent results relative to the other scoring schemes in the same experiment. This is due two factors: the variance in the results, and the fact that different scoring schemes are measuring different characteristics of the machine translation output. In Table 5, 6 of the 96 evaluations show negative results, but only one (NIST for Russian) is statistically significant. The NIST score has been shown to correlate with the adequacy of the translations [17]. A possible explanation of the low NIST score in this case is that for this language pair the proposed system has improved the fluency of the translations at the expense of adequacy because the parameters for the decoder and model interpolation parameters were tuned to maximize the BLEU score (which correlates well with fluency [17]).
Table 5  Performance results translating from a number of source languages into English.

| Source | BLEU | NIST  | WER  | PER  | GTM  | METEOR |
|--------|------|-------|------|------|------|--------|
| ar     | 0.4457 (0.00) | 8.9386 (0.00) | 0.4458 (0.00) | 0.3742 (0.00) | 0.7469 (0.00) | 0.6766 (0.00) |
| da     | 0.6640 (0.64) | 11.4500 (1.64) | 0.2560 (0.08) | 0.2174 (2.42) | 0.8338 (0.68) | 0.8154 (1.23) |
| de     | 0.6642 (0.79) | 11.4107 (0.44) | 0.2606 (2.18) | 0.2105 (0.14) | 0.8348 (-0.13) | 0.8132 (-0.07) |
| es     | 0.7345 (0.00) | 12.1384 (0.00) | 0.2117 (0.00) | 0.1668 (0.00) | 0.8519 (0.00) | 0.8541 (0.00) |
| fr     | 0.6666 (0.95) | 11.7443 (0.63) | 0.2548 (4.82) | 0.2172 (6.50) | 0.8408 (0.48) | 0.8293 (1.29) |
| id     | 0.5295 (9.56) | 10.3459 (4.11) | 0.3899 (21.17) | 0.3239 (4.65) | 0.7960 (1.35) | 0.7521 (2.35) |
| it     | 0.6702 (1.01) | 11.5604 (0.41) | 0.2590 (3.25) | 0.2090 (0.62) | 0.8351 (0.36) | 0.8171 (0.05) |
| ja     | 0.5971 (3.47) | 10.6346 (2.56) | 0.3779 (5.53) | 0.2842 (2.80) | 0.8125 (0.74) | 0.7669 (0.67) |
| ko     | 0.5898 (1.78) | 10.2151 (1.31) | 0.3891 (0.74) | 0.3138 (-0.10) | 0.7880 (0.36) | 0.7397 (0.35) |
| ms     | 0.5102 (10.19) | 9.9775 (2.75) | 0.4058 (18.53) | 0.3355 (3.59) | 0.7815 (0.18) | 0.7247 (2.49) |
| nl     | 0.6906 (2.55) | 11.9092 (1.47) | 0.2415 (3.21) | 0.1872 (1.73) | 0.8548 (0.39) | 0.8399 (0.36) |
| pt     | 0.6623 (0.35) | 11.6913 (0.26) | 0.2549 (2.52) | 0.2110 (2.68) | 0.8396 (0.02) | 0.8265 (-0.07) |
| ru     | 0.5877 (0.34) | 10.1233 (-1.10) | 0.3447 (1.99) | 0.2928 (1.71) | 0.7900 (0.15) | 0.7537 (-0.40) |
| th     | 0.4857 (1.50) | 9.5901 (1.17) | 0.4883 (-0.23) | 0.3579 (2.03) | 0.7608 (0.45) | 0.7104 (1.23) |
| vi     | 0.5118 (0.67) | 9.8588 (1.85) | 0.4274 (-0.05) | 0.3301 (0.12) | 0.7806 (1.05) | 0.7254 (0.43) |
| zh     | 0.5742 (0.00) | 10.1263 (0.00) | 0.3937 (0.00) | 0.3172 (0.00) | 0.7936 (0.00) | 0.7343 (0.00) |

Figures in parentheses are the percentage improvement in the score relative to the baseline score. Bold-bordered cells indicate those conditions where performance degraded. White cells indicate the proposed system’s performance is significantly different from the baseline (using 2000-sample bootstrap resampling with a 95% confidence level). TER scores were not tested for significance due to technical difficulties. ar, es and zh were also omitted since the systems were identical to the baseline. A key to the language abbreviations is given in Table 1.

Some of the language pairs showed striking improvements, in particular both of the Malay languages id and ms improved by over 3.5 BLEU points each using our technique. Interestingly Dutch, a relative of Malay, also improved substantially. This evidence points to a linguistic explanation for the gains. Malay has very simple and regular question structure, the question words appear at the front of question sentences (in the same way as the target language) and do not take any other function in the language (unlike the English do for example). Perhaps this simplicity of expression allowed our class-specific models to model the data well in spite of the reduced data caused by dividing the data. Another factor might be the performance of the classifier which was high for all these languages (around 98%). Unfortunately, it is hard to know the reasons behind the variety of scores in the table. One large factor is likely to be differences in corpus quality, and also the relationship between the source and target corpus. Some corpora are direct translations of each other, whereas others are translated through another language. Chinese was one such language, and this
Table 6 Performance results in terms of the BLEU score, comparing our proposed method with other techniques.

| Source | Baseline | No Classifier | Hard | Proposed |
|--------|----------|---------------|------|----------|
| ar     | 0.4457 (0.00) | 0.4457 (0.00) | 0.4457 (0.00) | 0.4457 |
| da     | 0.6598 (0.64) | 0.6647 (-0.11) | 0.6591 (0.74) | 0.664 |
| de     | 0.6590 (0.79) | 0.6651 (-0.14) | 0.6634 (0.12) | 0.6642 |
| es     | 0.7345 (0.00) | 0.7345 (0.00) | 0.7345 (0.00) | 0.7345 |
| fr     | 0.6603 (0.95) | 0.6594 (1.09) | 0.6605 (0.92) | 0.6666 |
| id     | 0.4833 (9.56) | 0.5029 (5.29) | 0.5276 (0.36) | 0.5295 |
| it     | 0.6635 (1.01) | 0.6660 (0.63) | 0.6644 (0.87) | 0.6702 |
| ja     | 0.5771 (3.47) | 0.5796 (3.02) | 0.5667 (5.36) | 0.5971 |
| ko     | 0.5795 (1.78) | 0.5837 (1.05) | 0.5922 (-0.41) | 0.5898 |
| ms     | 0.4630 (10.19) | 0.5015 (1.73) | 0.5057 (0.89) | 0.5102 |
| nl     | 0.6734 (2.55) | 0.6902 (0.06) | 0.6879 (0.39) | 0.6906 |
| pt     | 0.6600 (0.35) | 0.6643 (-0.30) | 0.6598 (0.38) | 0.6623 |
| ru     | 0.5857 (0.34) | 0.5885 (-0.14) | 0.5844 (0.56) | 0.5877 |
| th     | 0.4785 (1.50) | 0.4815 (0.87) | 0.4831 (0.54) | 0.4857 |
| vi     | 0.5084 (0.67) | 0.5095 (0.45) | 0.5041 (1.53) | 0.5118 |
| zh     | 0.5742 (0.00) | 0.5742 (0.00) | 0.5742 (0.00) | 0.5742 |

The column labeled No Classifier, is the same system as our proposed method, except that the classifier was not used. Instead a default model that assigned a class membership probability of 0.5 to each class for all sentences was used. The column labeled Hard corresponds to a system that used hard weights (either 1 or 0) for the class-dependent models. That is, the classifier was used, but the class membership was rounded up to 1 if the classifier gave a class membership probability > 0.5. The column labeled Proposed are the results from our proposed method. Figures in parentheses represent the percentage improvement of the proposed methods score relative to the alternative method. Cells with bold borders indicate those conditions where performance was degraded. A key to the language abbrevations is given in Table 1.

may explain why we were unable to improve on the baseline for this language even though we were very successful for both Japanese and Thai, which are relatives of Chinese.

The weight of the General SMT system in our experiments took values between 0.2 (for Indonesian) and 1.0 (for Chinese). The fact that this weight was never zero in our experiments indicates that a General SMT system trained on all the data is necessary.

6. Conclusion

In this paper we have presented a technique for combining all models from multiple SMT engines into a single decoding process. This technique allows for topic-dependent decoding with probabilistic soft weighting between the component models. We demonstrated the effectiveness of our approach on conversational data by building class-specific models to handle both questions and declarations. We carried out an extensive evaluation of the technique using a large number of language pairs and MT evaluation metrics. In most cases, we were able to show significant improvements over a system without model interpolation, and for some language pairs this approach excelled. The best improvement of all the language pairs was for Malaysian
(Malay)-English, which outperformed the baseline system by 4.7 BLEU points (from 0.463 to 0.510). In future research we would like to try the approach with larger sets of models as well as (possibly overlapping) subsets of the data produced using automatic clustering methods.

Acknowledgement

This is partly supported by Grant-in-Aid for Scientific Research (C) and Special Coordination Funds for Promoting Science and Technology of Ministry of Education, Culture, Sports, Science and Technology.

References

[1] R. Iyer and M. Ostendorf, “Modeling long distance dependence in language: Topic mixture versus dynamic cache models,” IEEE Trans. Speech Audio Process., vol.1, pp.236–239, 1994.
[2] D. Carter, “Improving language models by clustering training sentences,” Proc. ACL, pp.59–64, 1994.
[3] S. Hasan and H. Ney, “Clustered language models based on regular expressions for smt,” Proc. EAMT, Budapest, Hungary, 2005.
[4] H. Yamamoto and E. Sumita, “Bilingual cluster based models for statistical machine translation,” EMNLP-CoNLL-2007, pp.514–523, Prague, Czech Republic, June 2007.
[5] A. Finch, E. Denoual, H. Okuma, M. Paul, H. Yamamoto, K. Yasuda, R. Zhang, and E. Sumita, “The nict/atr speech translation system for iwslt 2007,” Proc. IWSLT, Trento, Italy, 2007.
[6] G. Foster and R. Kuhn, “Mixture-model adaptation for smt,” Proc. Second Workshop on Statistical Machine Translation, ACL, Prague, Czech Republic, June 2007.
[7] P. Koehn, “Pharaoh: A beam search decoder for phrase-based statistical machine translation models,” Machine translation: from real users to research: 6th conference of AMTA, Washington, DC, pp.115–124, 2004.
[8] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowa, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, “Moses: Open source toolkit for statistical machine translation,” ACL 2007: Proc. demo and poster sessions, pp.177–180, Prague, Czech Republic, June 2007.
[9] F.J. Och, “Minimum error rate training for statistical machine translation,” Proc. ACL, 2003.
[10] R. Rosenfeld, “A maximum entropy approach to adaptive statistical language modelling,” Computer Speech Lang., vol.10, pp.187–228, 1996.
[11] R. Lau, R. Rosenfeld, and S. Roukos, “Trigger-based language models: A maximum entropy approach,” Proc. International Conference on Acoustics, Speech and Signal Processing, pp.45–48, 1993.
[12] L. Zhang, “Maximum entropy modeling toolkit for python and c++,” 2004.
[13] G. Kikui, E. Sumita, T. Takezawa, and S. Yamamoto, “Creating corpora for speech-to-speech translation,” Proc. EUROspeech-03, pp.381–384, 2003.
[14] M. Paul, “Overview of the iwslt 2006 evaluation campaign,” Proc. IWLT, 2006.
[15] A. Stolcke, “Srilm - An extensible language model toolkit,” 1999.
[16] K. Papineni, S. Roukos, T. Ward, and W. Zhu, Bleu: A Method for Automatic Evaluation of Machine Translation, IBM Research Report rc22176 (w0109022), Thomas J. Watson Research Center, 2001.
[17] G. Doddington, “Automatic evaluation of machine translation quality using N-gram co-occurrence statistics,” Proc. HLT Conference, San Diego, California, 2002.
[18] S. Banerjee and A. Lavie, “Meteor: An automatic metric for mt evaluation with improved correlation with human judgments,” ACL-2005: Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pp.65–72, 2005.

Andrew Finch received the B.S. degree in mathematics and the M.Sc. degree in cognition, computing and psychology, both from the University of Warwick, England, in 1984 and 1990, respectively. He received a Ph.D. in computer science in 1995 from the University of York, England. From 1995 to 1997 he worked as a postdoctoral researcher in the computer vision research group at the University of York. In 1997 he received an Honorable Mention of the Pattern Recognition Society Award for Outstanding Contribution to the Pattern Recognition journal. From 1997 to the present day, he is a researcher at ATR Spoken Language Translation Research Laboratories, Kyoto, Japan. His current research interests include tagging, parsing, machine translation, and automatic paraphrasing.

Eiichiro Sumita received B.E. and M.E. degrees in computer science both from University of Electro-Communications, Japan, in 1980 and 1982 respectively. He received a Ph.D in engineering from Kyoto University, Japan, in 1999. He is head of the Language Translation group at NICT-ATR Research Laboratories, Kyoto, Japan. His research interests include natural language processing, machine translation, information retrieval, automatic evaluation, e-Learning and parallel processing. He is a member of the IPSJ and the ACL.

Satoshi Nakamura received the Ph.D. degree in information science from Kyoto University in 1992. He worked with ATR Interpreting Telephony Research Laboratories from 1986 to 1989. From 1994 to 2000, he was an Associate Professor at the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. In 1996, he was a Visiting Research Professor of the CAIP Center of Rutgers University in New Jersey, USA. He is currently the Head of the MASTAR Project at NICT-ATR, Japan. His current research interests include speech recognition, speech translation, spoken dialogue systems, stochastic modeling of speech, and microphone arrays. He received the Awaya award from the Acoustical Society of Japan in 1992, and the Interaction 2001 Best Paper Award from the Information Processing Society of Japan in 2001. He served as an Editor for the Journal of the IEICE Information from 2000 to 2002.