Two methods for stabilizing MERT: NICT at IWSLT 2009

Masao Utiyama, Hirofumi Yamamoto, Eiichiro Sumita

†MASTAR project, NICT
Hikaridai 2-2-2, Keihanna Science City, 619-0288 Kyoto, Japan
mutiyama@nict.go.jp

Abstract

This paper describes the NICT SMT system used in the International Workshop on Spoken Language Translation (IWSLT) 2009 evaluation campaign. We participated in the Challenge Task. Our system was based on a fairly common phrase-based machine translation system. We used two methods for stabilizing MERT.

1. Introduction

This paper describes the NICT SMT system used in the International Workshop on Spoken Language Translation (IWSLT) 2009 evaluation campaign. We participated in the Challenge Task. Our system was based on a fairly common phrase-based machine translation system [1], which was built within the framework of a feature-based exponential model. The model has the following features:

- Phrase translation probability form source to target
- Inverse phrase translation probability
- Lexical weighting probability from source to target
- Inverse lexical weighting probability
- Phrase penalty
- Language model probability
- Lexical reordering probability
- Simple distance-based distortion model
- Word penalty

The decoder can operate on the same principles as the MOSES decoder [2]. For the training of SMT models, we used a training toolkit adapted from the MOSES decoder. We used GIZA++ [3] for word alignment and SRILM [4] for language modeling. We used 4-gram language models trained with modified Kneser–Ney smoothing. The language models were trained with SMT training corpora on the target side. Minimum error rate training (MERT) was used to tune the decoder's parameters on the basis of the bilingual evaluation understudy (BLEU) score, and training was performed using the standard technique developed by Och [5].

2. Language resources

For the training data, we used `IWSLT09_BTEC.train.*`, `IWSLT09.devset*` and `IWSLT09_CT.train.*`. We expanded the devset data by pairing each source sentence with all of its reference translations. We made a phrase and reordering table from this training data. We also made a language model from the CT portion of this data and a language model from the remaining data. These models were combined log-linearly.

For the development data, we used part of the training data that were extracted by the method described in Section 4.1. The development data were excluded from the training data when we tuned parameters. However, the development data were added to the training data when we translated the development test data and test data.

For the development test data, we used `IWSLT09_CT.devset.*.with_interpreter.txt`.

We used in-house tokenizers to tokenize Chinese and English sentences. In making our models, we lowercased English sentences in the English-Chinese (EC) translation, but we didn't lowercased English sentences in the Chinese-English (CE) translation.

3. Combination of Chinese segmentations

We combined two kinds of Chinese segmentations when we made our phrase and reordering models. Although we did not have enough time to investigate the stability of this method, it improved BLEU scores about 0.5 to 1 points based on our limited experiments. After the submission, we confirmed that this method slightly improved BLEU scores for the test set.

In the first segmentation method, we segmented Chinese texts into words using our in-house tokenizer and made a phrase (reordering) table. (The experiments in Sections 4.1 and 4.2 used this method.)

We then segmented Chinese texts into characters with '$\langle w \rangle$' tags inserted between words. An example is "$\langle w \rangle\ c_1\ c_2\ \langle w \rangle\ c_3\ \langle w \rangle\ c_4\ c_5\ c_6$". We also inserted '$\langle w \rangle$' into English texts. Next, we made a phrase (reordering) table from this data.

Finally, we combined these two phrase (reordering) tables by segmenting the phrases in the first phrase (reordering)

table into characters.

When we translated a Chinese sentence into an English sentence, we first segmented the Chinese sentence into words then applied the second segmentation method. When we translated an English sentence into a Chinese sentence, we removed "$\langle w \rangle$" from the output.

4. Two methods for stabilizing MERT

4.1. Devset sampling

The first method we used was to extract development data from the training data that were similar to the input texts. We call this method "devset sampling." Note that the sampled sentences were excluded from the training data when training and tuning the model initially in order to avoid over-fitting problems. After the initial training and tuning, we added the sampled sentences again into the training data to train the final model whose parameters were the ones tuned by the sampled sentences.

For each sentence in the development test data, we extracted the most similar 100 sentences from the training data. We used the average of BLEU1, BLEU2, BLEU3, and BLEU4 scores as the similarity score. This score was calculated from each input sentence and the foreign part of a bi-sentence. The input sentence was regarded as the reference when we calculated BLEU scores. We used several most similar sentences from the 100 similar sentences of each sentence to make 1000 sentence development data. This means that if the number of test sentences was 500, we used the most similar 2 sentences of each test sentence to make 1000 sentence development data. See Appendix A for examples.

We used this development data to tune parameters. We run MERT ten times on this development data to calculate the average BLEU score for the development test data. The average BLEU scores were 32.16 and 28.66 for EC and CE, respectively. In contrast, when we tuned parameters on a randomly sampled 1000 sentence development data, the average BLEU scores were 30.34 and 26.12 for EC and CE, respectively.

4.2. Averaged MERT

The second method we used was to run MERT several times on a development data, then average tuned parameters to get final parameters. We call this method "averaged MERT".

In order to understand why this method is reasonable, recall that the score of an English sentence $\mathbf{e}$ w.r.t. an input sentence $\mathbf{f}$ is

$$\sum_{m=1}^{M} \lambda_m h_m(\mathbf{e}, \mathbf{f})$$

where $h_m(\mathbf{e}, \mathbf{f})$ is a feature function and λ_m is a weight. Calculating the average of parameters (weights) means that we calculate the average of scores obtained by using different parameters. Consequently, using the averaged parameters is a kind of using a system combination method.

When we run MERT ten times on the development data described in the previous section and averaged the parameters, the BLEU scores for the development test data were 32.61 and 29.24 for EC and CE, respectively. In contrast, when we used the parameters that obtained the maximum BLEU score on the development data, the BLEU scores were 31.93 and 28.49 for EC and CE, respectively. See Appendix B for additional experiments on averaged MERT.

Using these two methods, the BLEU scores were improved from 30.34 and 26.12 to 32.61 and 29.24 for EC and CE, respectively.

5. Official results

The BLEU scores for our official submissions are shown in Table 1. In this table, "c+p" and "nc+np" mean "case+punc" and "nocase+nopunc", respectively. We used 1-best sentences to translate ASR outputs.

Table 1: BLEU scores for our official submissions

	EC		CE	
	c+p	nc+np	c+p	nc+np
ASR	35.83	35.44	26.67	25.80
CSR	38.42	38.15	29.70	28.72

6. What we tried but didn't work

This section describes several methods that we tried but didn't work. We used the original Chinese segmentations in these experiments.

6.1. Increasing the size of the CT corpus

In this method, we added several replications of each sentence of the CT corpus when we added them to the BTEC corpus. When we made the size of the CT corpus to be about the same size of the BTEC corpus, the BLEU scores were reduced from 31.25 to 30.89 and from 26.11 to 25.12 for EC and CE, respectively. (The baseline BLEU scores were different from those in other experiments, because this comparison was done in an early stage of our system development.)

6.2. Alignment with lowercased prefixes

In this method, we used the lowercased 4-letter prefixes of English words in word alignment. When we applied this method, the BLEU scores were reduced from 32.22 to 29.58 for EC and from 26.91 to 26.73 for CE.

6.3. Replacing numbers with a special symbol

We replaced each occurrence of numbers in the corpus with a special symbol and trained an SMT system. That is, all numbers in the corpus were replaced with the same special symbol. If an input sentence for the SMT system had a num-

ber, the number was replaced with the special symbol before inputing it to the SMT system. The special symbol in the output sentence was replaced with the translation of the number in the input sentence.

We tried this method because the CT corpus contained many numbers. However, this method didn't work very well. One reason is that the CT corpus contained idiomatic expressions that were not able to be handled with this method. For example, a Chinese word sequence "0 0 0" is translated into "triple o". The BLEU scores were reduced from 32.22 to 29.56 and from 26.91 to 24.17 for EC and CE, respectively.

7. Conclusions

We participated in the Challenge Task. Our system was based on a fairly common phrase-based machine translation system. We used two methods for stabilizing MERT. Both methods improved BLEU scores.

8. References

[1] A. Finch and E. Sumita, "Dynamic model interpolation for statistical machine translation," in *Proceedings of the Third Workshop on Statistical Machine Translation*, 2008, pp. 208–215.

[2] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, 2007, pp. 177–180.

[3] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[4] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *ICSLP*, 2002.

[5] F. J. Och, "Minimum error rate training in statistical machine translation," in *ACL*, 2003.

[6] C. S. Fordyce, "Overview of the IWSLT 2007 evaluation campaign," in *IWSLT*, 2007.

A. Sentences selected by devset sampling

We list the test sentences and the most similar sentences selected by the method described in Section 4.1. The format is "test sentence // similar sentence".

- hotel royal plaza may i help you // holiday inn crowne plaza may i help you
- yes can you tell me the number of people type of room and approximate budget please // yes would you tell me the address and phone number of the hotel please
- yes let me check for vacancies // yes let me check hold on a moment please
- sorry to keep you waiting // sorry to keep you waiting
- we have two types of rooms available in your budget range // we have two types of dressing japanese or french
- we have a standard twin room at one hundred and forty - five to one hundred and fifty dollars // just now we have rooms that cost one hundred and forty dollars a night
- we have a superior twin room at one hundred and fifty - seven dollars to one hundred and seventy dollars // okay we have a single room with bath for fifty - seven dollars a night
- thank you very much can i have your name and contact number please // can i have your name and a contact number please
- yes go ahead // yes go ahead
- yes we do offer a special breakfast menu for children in addition to our standard menu // thank you for waiting yes we do have a japanese room available if you like
- our special breakfast for children includes pancakes milk and fruit for ten dollars // okay continental breakfast and that will be ten dollars per night
- and what will your method of payment be for the room ms suzuki // and what will the method of payment be
- that 'll be fine can i have your mastercard number and the expiration date please // okay that 's fine could i have your visa number and the expiration date please
- thank you let me just repeat that // thank you let me just repeat that
- mastercard number five two seven nine three nine two o two four six nine zero zero nine eight // number five two seven nine three nine two zero two four six nine zero zero nine eight correct
- the expiration date is april nineteen ninety - six // the expiration date is april nineteen ninety - six
- and when is your expected time of arrival // and what is your expected time of arrival on tuesday the twenty - fifth
- okay thank you very much we 'll be waiting for you on october twenty - seventh at about seven p m // okay then we 'll be waiting for you on thursday october twenty - seventh thank you very much

- thank you for making a reservation at the hotel royal plaza // thank you for making a reservation at the new washington hotel
- the milburn may i help you // the milburn may i help you
- yes miss suzuki when would you like a reservation for // then when would you like to make a reservation for

B. Additional experiments on averaged MERT

We used the development data for the IWSLT-2007 Japanese-English translation task [6] to verify the usefulness of averaged MERT. The development data consisted of five sets, devset1, devset2, devset3, devset4, and devset5. Each of these data sets had about 500 sentences. The numbers of reference translations were 16 for devset1, devset2, and devset3 and 7 for devset4 and devset5.

We used devset1 to tune our SMT system and used devset2, devset3, devset4, and devset5 as the testsets to evaluate the performance of our SMT system in terms of BLEU scores. Hereafter, we refer to devset2, ..., devset5 as set2, ..., set5, respectively.

We used a bootstrap method. First, we run MERT 100 times on devset1 using different random start points. Consequently, we obtained 100 parameter sets. From these parameter sets, we calculated the averages and standard deviations of BLEU scores for 1, 2, 3, 5, 7, 10, 20, 30, 50, 70, and 100 parameter sets by sampling these parameter sets. Our sampling schemes were with replacement and we sampled 100 parameter sets for each number of parameter sets.

We compared two methods for combining parameter sets. In the first method, we used the averaged parameters. In the second method, we used the parameters that obtained the maximum BLEU score on devset1.

The BLEU scores for set2, set3, set4, and set5 are shown in Tables 2, 3, 4, and 5.

In these tables, the numbers in "No." columns represent the number of parameter sets. The labels in the top rows, "average" and "maximum", are the methods we used. The figures in "av." and "(std.)" columns are the averages and standard deviations of BLEU scores for 100 samples of parameter sets.

These tables show that we obtained higher BLEU scores when we used more parameter sets. In addition, the "average" method was better than the "maximum" method in all cases (excepting the case where the number of parameter sets was 1 in which these two methods were identical).

We conducted similar experiments on the IWSLT-2008 Chinese-English translation task data and observed similar results.

From these experiments, we concluded that averaged MERT improve BLEU scores.

Table 2: BLEU scores for set2

method	average	maximum
No.	av. (std.)	av. (std.)
1	62.22 (0.54)	62.22 (0.54)
2	62.59 (0.41)	62.32 (0.42)
3	62.63 (0.37)	62.08 (0.59)
5	62.72 (0.38)	62.18 (0.53)
7	62.72 (0.29)	62.14 (0.56)
10	62.73 (0.27)	62.14 (0.54)
20	62.71 (0.21)	62.27 (0.52)
30	62.73 (0.21)	62.16 (0.55)
50	62.69 (0.19)	62.36 (0.45)
70	62.70 (0.16)	62.42 (0.41)
100	62.71 (0.15)	62.50 (0.33)

Table 3: BLEU scores for set3

method	average	maximum
No.	av. (std.)	av. (std.)
1	61.12 (0.53)	61.12 (0.53)
2	61.51 (0.51)	61.08 (0.44)
3	61.59 (0.46)	61.07 (0.54)
5	61.67 (0.38)	61.13 (0.50)
7	61.72 (0.34)	61.11 (0.47)
10	61.71 (0.33)	61.32 (0.50)
20	61.65 (0.29)	61.35 (0.41)
30	61.67 (0.26)	61.34 (0.43)
50	61.59 (0.20)	61.45 (0.39)
70	61.62 (0.24)	61.41 (0.33)
100	61.64 (0.21)	61.46 (0.28)

Table 4: BLEU scores for set4

method	average	maximum
No.	av. (std.)	av. (std.)
1	26.24 (0.79)	26.24 (0.79)
2	26.67 (0.61)	26.36 (0.76)
3	26.74 (0.51)	26.46 (0.86)
5	26.83 (0.49)	26.33 (0.68)
7	27.02 (0.44)	26.24 (0.78)
10	26.93 (0.36)	26.15 (0.65)
20	27.03 (0.35)	26.22 (0.64)
30	27.04 (0.26)	26.03 (0.29)
50	27.03 (0.23)	26.04 (0.42)
70	27.02 (0.22)	26.03 (0.31)
100	26.99 (0.22)	26.11 (0.21)

Table 5: BLEU scores for set5

method	average	maximum
No.	av. (std.)	av. (std.)
1	21.20 (0.63)	21.20 (0.63)
2	21.45 (0.49)	21.31 (0.57)
3	21.53 (0.44)	21.35 (0.58)
5	21.57 (0.43)	21.28 (0.51)
7	21.73 (0.37)	21.33 (0.52)
10	21.70 (0.33)	21.25 (0.42)
20	21.83 (0.29)	21.23 (0.49)
30	21.88 (0.29)	21.20 (0.36)
50	21.84 (0.24)	21.27 (0.37)
70	21.84 (0.22)	21.24 (0.36)
100	21.87 (0.21)	21.31 (0.31)