Description of the JHU System Combination Scheme for WMT 2011

Daguang Xu
Johns Hopkins University
Baltimore, USA
dxu5@jhu.edu

Yuan Cao
Johns Hopkins University
Baltimore, USA
yuan.cao@jhu.edu

Damianos Karakos
Johns Hopkins University
Baltimore, USA
damianos@jhu.edu

Abstract

This paper describes the JHU system combination scheme used in WMT-11. The JHU system combination is based on confusion network alignment, and inherited the framework developed by (Karakos et al., 2008). We improved our core system combination algorithm by making use of TER-plus, which was originally designed for string alignment, for alignment of confusion networks. Experimental results on French-English, German-English, Czech-English and Spanish-English combination tasks show significant improvements on BLEU and TER by up to 2 points on average, compared to the best individual system output, and improvements compared with the results produced by ITG which we used in WMT-10.

1 Introduction

System combination aims to improve the translation quality by combining the outputs from multiple individual MT systems. The state-of-the-art system combination methodologies can be roughly categorized as follows (Karakos et al., 2010):

1. *Confusion network based:* confusion network is a form of lattice with the constraint that all paths need to pass through all nodes. An example of a confusion network is shown in Figure 1.

   Here, the set of arcs between two consecutive nodes represents a bin, the number following a word is the count of this word in its bin, and each bin has the same size. The basic methodology of system combination based on confusion network includes the following steps: (a) Choose one system output as the "skeleton", which roughly decides the word order. (b) Align further system outputs to the skeleton, thus forming a confusion network. (c) Rescore the final confusion network using a language model, then pick the best path as the output of combination.

   A textual representation (where each line contains the words and counts of each bin) is usually the most convenient for machine processing.

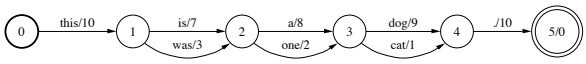

Figure 1: *Example confusion network. The total count in each bin is 10.*

2. *Joint optimization based:* unlike building confusion network, this method considers all system outputs at once instead of incrementally. Then a log-linear model is used to derive costs, followed by a search algorithm to explore the combination space (Jayaraman et al., 2005; Heafield et al., 2009; He et al., 2009).

3. *Hypothesis selection based:* this method only includes algorithms that output one of the input translations, and no word selection from multiple systems is performed. Typical algorithms can be found in (Rosti et al., 2007).

Proceedings of the 6th Workshop on Statistical Machine Translation, pages 171–176,
Edinburgh, Scotland, UK, July 30–31, 2011.

This paper describes the JHU system combination submitted to the Sixth Workshop on Statistical Machine Translation (WMT-11) (http://statmt.org/wmt11/index.html). The JHU system combination is confusion network based as described above, following the basic system combination framework described in (Karakos et al., 2008). However, instead of ITG alignments that were used in (Karakos et al., 2008), alignments based on TER-plus (Snover et al., 2009) were used now as the core system alignment algorithm.

The rest of the paper is organized as follows: Section 2 introduces the application of TER-plus in system combination. Section 3 introduces the JHU system combination pipeline. Section 4 presents the combination results and concluding remarks appear in Section 5.

2 Word Reordering for Hypothesis Alignment

Given the outputs of multiple MT systems, we would like to reorder and align the words of different hypothesis in a way such that an objective function is optimized, thus reaching better translations by making use of more information. In our system combination scheme, the objective function was based on Translation-Edit-Rate Plus (TER-plus).

2.1 Introduction to TER-plus

TER-plus is an extension of Translation Error Rate (TER) (Snover et al., 2006). TER is an evaluation metric for machine translation; it generalizes Word Error Rate (WER) by allowing block shifts in addition to the edit distance operations. However, one problem with TER is that only exact match of word blocks are allowed for shifting; this constraint might be too strict as it sometimes prevents reasonable shifts if two blocks have similar meanings.

TER-plus remedies this problem by introducing new flexible matches between words, thus allowing word substitutions and block shifts with costs much lower than that of TER. Specifically, substitution costs are now dependent on whether the words have the same stem (stem matches) or are synonyms (synonym matches). These operations relax the shifting constraints of TER; shifts are now allowed if the words of one string are synonyms or share the same stem as the words of the string they are compared to (Snover et al., 2009).

TER-plus identifies words with the same stem using the Porter stemming algorithm (Porter et al., 1980), and identifies synonyms using the WordNet database (Miller et al., 1995).

2.2 TER-plus for system combination

Originally, TER-plus was designed for aligning together word strings. However, similar to the work of (Karakos et al., 2010), who extended ITG to allow bilingual parsing of two *confusion networks* (by treating each confusion network bin as a multi-word entity), we converted the basic TER-plus code to take into account multiple words present in confusion network bins. Specifically, we define the cost of aligning two confusion network bins as (Karakos et al., 2010)

$$cost(b_1, b_2) = \frac{1}{|b_1||b_2|} \sum_{w_1 \in b_1} \sum_{w_2 \in b_2} \mathcal{C}(w_1, w_2)$$

in which b_1,b_2 are the confusion network bins which are candidates for alignment, $|\cdot|$ is the size of a bin, w_1, w_2 are words in b_1 and b_2 respectively, and $\mathcal{C}(w_1, w_2)$ is defined as follows:

$$\mathcal{C}(w_1, w_2) = \begin{cases} 0 & w_1 \text{ matches } w_2 \\ 0.5 & w_2 \text{ is deleted} \\ 0.6 & w_2 \text{ is inserted} \\ 0.2 & w_1 \text{ and } w_2 \text{ are synonyms} \\ 0.2 & w_1 \text{ and } w_2 \text{ share stems} \\ 1 & \text{none of the above} \end{cases}$$

Furthermore, the bin shift cost is set to 1.5. These numbers are empirically determined based on experimental results.

Similar to (Karakos et al., 2010), when a bin gets "deleted", it gets replaced with a *NULL* arc, which simply encodes the empty string, and is otherwise treated as a regular token in the alignments.

3 The JHU System Combination Pipeline

We now describe the JHU system combination pipeline in which TER-plus is used as the core confusion network alignment algorithm as introduced in the previous section.

3.1 Combination procedure overview

The JHU system combination scheme is based on confusion network as introduced in section 1. The confusion networks are built in two stages:

1. **Within-system combination:** (optional, only applicable in the case where per-system n-best lists are available.) the within-system combination generates system-specific confusion networks based on the alignment of the n-best translations.

2. **Between-system combination:** incremental alignment of the confusion networks of different systems generated in step 1, starting from 2-system combination up to the combination of all systems. The order with which the systems are selected is based on the individual BLEU scores (i.e., the best two systems are first combined, then the 3rd best is aligned to the resulting confusion network, etc.)

For the between-system combination we made use of TER-plus as described in section 2.2.

3.2 Language model Rescoring with Finite-State Transducer Operations

Once the between-system confusion networks are ready (one confusion network per sentence), a path through each of them has to be selected as the combination output. In order to pick out the the most fluent word sequence as the final translation, we need to rescore the confusion networks using a language model. This task can be performed efficiently via finite state transducer (FST) operations (Allauzen et al., 2002). First, we build an FST for each confusion network, called CN-FST. Since the confusion network is just a sequence of bins and each bin is a superposition of single words, the CN-FST can be built as a linear FST in a straightforward way (see Figure 1).

A 5-gram language model FST (LM-FST) is then built for each sentence. To build the LM-FST, we refer to the methodology described in (Allauzen et al., 2003). In brief, the LM-FST is constructed in the following way:

1. Extract the vocabulary of each segment.

2. Each state of the FST encodes an n-gram history ($n-1$ words). Each (non-null) arc that originates from that state corresponds uniquely to a word type (i.e., word that follows that history in the training data).

3. The cost of each word arc is the corresponding language model score (negative log-probability, based on the modified Kneser-Ney formula (Kneser, 1995) for that n-gram).

4. Extra arcs are added for backing-off to lower-order histories, thus allowing all possible word strings to receive a non-zero probability.

In order to deal with the situation where a word in the confusion network is not in the vocabulary of the language model, we need to build another simple transducer, namely, the "unknown word" FST (UNK-FST), to map this word to the symbol *<unk>* that encodes the out-of-vocabulary (OOV) words. Note that this is useful only if one builds *open-vocabulary language models* which always give a non-zero probability to OOV words; e.g., check out the option *-unk* of the SRILM toolkit (Stolcke, 2002). (Obviously, the UNK-FST leaves all other words unmodified.)

After all these three transducers have been built, they are composed in the following manner (for each sentence):

CN-FST .o. UNK-FST .o. LM-FST

Note that a possible *re-weighting* of the arc costs of the CN-FST can be done in order to better account for the different dynamic ranges between the CN costs and the LM-FST costs. Furthermore, to avoid too many word deletions (especially in regions of the confusion network where the words disagree most) an additive *word deletion penalty* can be added to all *NULL* arcs. The best (minimum-cost) path from this resulting FST is selected as the output translation of the system combination for that sentence.

3.3 System combination pipeline summary

We now summarize the JHU system combination end-to-end pipeline as follows(since BLEU score is a key metric in the WMT11 translation evaluation, we use BLEU score as the system ranking criteria. The BLEU score we computed for the experiments below are all case-insensitive):

1. Process and re-format (lowercase, tokenize, romanize, etc.) all individual system outputs. Note that we compute the case-insensitive BLEU score in our experiments.

2. Build LM-FST and UNK-FST for each sentence.

3. Decide the between-system combination order according to the 1-best output BLEU score of individual systems.

4. Do between-system combination based on the order decided in step 3 using TER-plus.

5. Rescore the confusion network and start tuning on the parameters: convert the between-system confusion network into FST, compose it with the UNK-FST and with the LM-FST. When composing with LM-FST, try different CN arc coefficients (we tried the range $\{5, \ldots, 21\}$), and unknown word insertion penalties (we tried the values $\{0.3, 0.5, 0.7, 1\}$).

6. Compute the BLEU score for all *m*-syst_*x*_*y* outputs, where *m* is the number of systems for combination, *x* is the weight and *y* is the insertion penalty.

7. Among all the scores computed in step 6, find the best BLEU score, and keep the corresponding parameter setting(*m*, *x*, *y*).

8. Apply the best parameter setting to the test dataset for evaluation.

Obviously, if n-best outputs from systems are available, an extra step of producing within-system combinations (and searching for the best n-best size) will also be executed.

4 Results

In WMT11, we participated in French-English, German-English, Czech-English and Spanish-English system combination tasks. Although we followed the general system combination pipeline introduced in 3.3, we did not do the within-system combination since we received only 1-best outputs from all systems.

We built both primary and contrastive systems, and they differ in the way the 5-gram language models were trained. The language model for the primary system was trained with the monolingual Europarl, news commentary and news crawl corpus provided by WMT11. The language model for the contrastive system was trained using only the 1-best outputs from all individual systems (sentence-specific language model).

The number of systems used for combination tuning in each language pair was: 24 for French-English, 26 for German-English, 12 for Czech-English, and 16 for Spanish-English. The best results for the combination in the primary system made use of 23 systems for French-English, 5 systems for German-English, 10 systems for Czech-English, 10 systems for Spanish-English. In the contrastive system, the number of systems were 20, 5, 6, 10 respectively.

The TER and BLEU scores on the development set for the best individual system, the primary and contrastive combinations are given in Table 1, and the scores for test set are given in Table 2. From the results we see that, compared with the best individual system outputs, system combination results in significantly improved BLEU scores and remarkable reductions on TER, for all language pairs. Moreover, we observe that the primary system performs slightly better than the contrastive system in most cases.

We also did the experiment of xx-English which made combinations of all English outputs available across different source languages. We used 35 systems in this experiment for both primary and contrastive combination, and best result made use of 15 and 16 systems respectively. The development and test set results are shown in the "xx-en" column in table 1 and 2 respectively. From the results we see the improvements on TER and BLEU scores of both development and test sets almost doubled compared with the best results of single language pairs.

To make a comparison with the old technique we used in WMT10 system combination task, we ran the WMT11 system combination task using ITG with surface matching. The detailed implementation is described in (Narsale, 2010). Table 3 and 4 show the WMT11 results using ITG for alignment respectively. It can be seen that TER-plus outperforms ITG

System	fr-en		de-en		cz-en		es-en		xx-en	
	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU
Best single system	56.2	28.1	60.1	23.6	**54.9**	27.9	51.8	30.2	51.8	30.2
Primary combination	**49.2**	**32.6**	**58.1**	**25.7**	55.1	28.7	**48.3**	**33.7**	**44.9**	35.5
Contrastive combination	49.8	32.3	58.2	25.6	**54.9**	**28.9**	49.1	33.3	45.0	**37.2**

Table 1: Results for all language pairs on development set. The best number in each column is shown in **bold**.

System	fr-en		de-en		cz-en		es-en		xx-en	
	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU
Best single system	58.2	30.5	65.1	23.5	**59.7**	29.1	60.0	28.9	58.2	30.5
Primary combination	**55.9**	**31.9**	**64.4**	**25.0**	60.1	29.6	**55.4**	**33.5**	**51.7**	36.3
Contrastive combination	56.5	31.6	65.7	24.4	59.9	**29.8**	56.5	33.4	52.5	**36.5**

Table 2: Results for all language pairs on test set. The best number in each column is shown in **bold**.

System	fr-en		de-en		cz-en		es-en		xx-en	
	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU
Best single system	56.2	28.1	60.1	23.6	54.9	27.9	51.8	30.2	51.8	30.2
Primary combination	**49.0**	**32.5**	**57.6**	**25.0**	**54.6**	**28.1**	**48.8**	**33.1**	**45.3**	35.7
Contrastive combination	56.1	31.7	58.0	24.9	55.0	28.0	49.4	33.0	45.6	**35.9**

Table 3: Results for all language pairs on development set using ITG. The best number in each column is shown in **bold**.

System	fr-en		de-en		cz-en		es-en		xx-en	
	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU
Best single system	58.2	30.5	65.1	23.5	**59.7**	29.1	60.0	28.9	58.2	30.5
Primary combination	**55.9**	**31.9**	**64.5**	**24.7**	60.1	29.4	**55.8**	**33.0**	**52.2**	35.0
Contrastive combination	56.6	31.4	64.7	24.4	60.7	**29.6**	56.6	**33.0**	52.9	**35.3**

Table 4: Results for all language pairs on test set using ITG. The best number in each column is shown in **bold**.

almost in all results. We will experiment with ITG and *flexible match costs* and will report results in a subsequent publication.

5 Conclusion

We described the JHU system combination scheme that was used in WMT-11. The JHU system combination system is confusion network based, and we demonstrated the successful application of TERplus (which was originally designed for string alignment) to confusion network alignment. The WMT-11 submission results show that significant improvements on the TER and BLEU scores (over the best individual system) were achieved.

Acknowledgments

This work was supported by the DARPA GALE program Grant No HR0022-06-2-0001. We would also like to thank the IBM Rosetta team for their strong support in the system combination evaluation tasks.

References

D. Karakos, J. Smith, and S. Khudanpur. 2010. *Hypothesis ranking and two-pass approaches for machine translation system combination*. Acoustics Speech and Signal Processing (ICASSP), IEEE International Conference on.

S. Jayaraman and A. Lavie. 2005. *Multi-engine machine translation guided by explicit word matching*. Proc. EAMT:143–152.

K. Heafield, G. Hanneman, and A. Lavie. 2009. *Machinetranslation system combination with flexible word ordering*. Proc. EACL 2009, WSMT.

X. He and K. Toutanova. 2009. *Joint optimization for machine translation system combination*. Proc. EMNLP.

A.-V.I. Rosti, S. Matsoukas, and R. Schwartz. 2007. *Improved word-level system combination for machine translation*. Proceedings of Association for Computational Linguistics(ACL)

D. Karakos, J. Eisner, S. Khudanpur, M. Dreyer. 2008. *Machine translation system combination using ITG-based alignments*. Proceedings of Association for Computational Linguistics(ACL) HLT, Short Papers (Companion Volume):81-84.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, J. Makhoul. 2006 *A Study of Translation Edit Rate with Targeted Human Annotation*. Proceedings of Association for Machine Translation in the Americas.

G.Miller. 1995 *WordNet: A Lexical Database for English.* . Communications of the ACM Vol. 38, No. 11.

M. Snover, N. Madnani, B. Dorr, R. Schwartz. 2009 *Fluency, adequacy, or HTER? Exploring different human judgments with a tunable MT metric*. Proceedings of the Fourth Workshop on Statistical Machine Translation at the 12th Meeting of the European Chapter of the Association for Computational Linguistics (EACL-2009), Athens, Greece.

M.F.Porter. 1980 *An algorithm for suffix stripping*. Program 14(3):130-137

C. Allauzen, M. Mohri, B. Roark 1980 *Generalized Algorithms for Constructing Statistical Language Models*. Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, July 2003, pp. 40-47.

Sushant Narsale. 2010 *JHU system combination scheme for WMT 2010*. Proceedings of Fifth Workshop on Machine Translation, ACL.

R. Kneser, Ney. H. 2010 *Improved backing-off for m-gram language modeling*. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP.

A. Stolcke 2002 *SRILM - An Extensible Language Modeling Toolkit*. Proceedings of International Conference on Spoken Language Processing.

C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, Mehryar Mohri. 2002 OpenFst: A General and Efficient Weighted Finite-State Transducer Library Proceedings of the Ninth International Conference on Implementation and Application of Automata, (CIAA 2007), vol. 4783, Lecture Notes in Computer Science, pages 11-23, 2007

WMT11 official webpage. *http://statmt.org/wmt11 /index.html*