JHU System Combination Scheme for WMT 2010

Sushant Narsale
Johns Hopkins University
Baltimore, USA.
sushant@jhu.edu

Abstract

This paper describes the JHU system combination scheme that was used in the WMT 2010 submission. The incremental alignment scheme of (Karakos et.al, 2008) was used for confusion network generation. The system order in the alignment of each sentence was learned using SVMs, following the work of (Karakos et.al, 2010). Additionally, web-scale n-grams from the Google corpus were used to build language models that improved the quality of the combination output. Experiments in Spanish-English, French-English, German-English and Czech-English language pairs were conducted, and the results show approximately 1 BLEU point and 2 TER points improvement over the best individual system.

1 Introduction

System Combination refers to the method of combining output of multiple MT systems, to produce a output better than each individual system. Currently, there are several approaches to machine translation which can be classified as phrase-based, hierarchical, syntax-based (Hildebrand and Vogel, 2008) which are equally good in their translation quality even though the underlying frameworks are completely different. The motivation behind System Combination arises from this diversity in the state-of-art MT systems, which suggests that systems with different paradigms make different errors, and can be made better by combining their strengths.

One approach of combining translations is based on representing translations by confusion network and then aligning these confusion networks using string alignment algorithms (Rosti et.al, 2009), (Karakos and Khudanpur, 2008). Another approach generates features for every translation to train algorithms for ranking systems based on their quality and the top ranking output is considered to be a candidate translation, (Hildebrand and Vogel, 2008) is an example of ranking based combination. We use ideas from ranking based approaches to learn order in which systems should be aligned in a confusion network based approach.

Our approach is based on incremental alignment of confusion networks (Karakos et.al, 2008), wherein each system output is represented by a confusion network. The confusion networks are then aligned in a pre-defined order to generate a combination output. This paper contributes two enhancements to (Karakos et.al, 2008). First, use of Support Vector Machines to learn order in which the system outputs should be aligned. Second, we explore use of Google n-grams for building dynamic language model and interpolate the resulting language model with a large static language model for rescoring of system combination outputs.

The rest of the paper is organized as follows: Section 2 illustrates the idea and pipeline of the baseline combination system; Section 3 gives details of SVM ranking for learning system order for combination; Section 4 explains use of Google n-gram based language models; Results are discussed in Section 5; Concluding remarks are given in Section 6;

2 Baseline System Combination

This section summarizes the algorithm for baseline combination. The baseline combination pipeline includes three stages:

1. Representing translations by confusion networks.

Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR, pages 311–314,
Uppsala, Sweden, 15-16 July 2010.

2. Generating between system confusion networks.

3. Rescoring the final confusion network.

Confusion networks are compressed form of lattices with a constraint that all paths should pass through all nodes. Each system output is represented by an equivalent confusion network. The per-system confusion networks are aligned one at a time. The order in which systems are aligned is usually decided by evaluation of system's performance. Two alternatives for deciding the system order are discussed in Section 3. Inversion-Transduction Grammar (Wu, 1997) is used for alignments and the cost function for aligning two confusion networks is

$$cost(b_1, b_2) = \frac{1}{|b_1||b_2|} \sum_{w \in b_1} \sum_{v \in b_2} c(v)c(w)\mathbf{1}(w \neq v)$$

where b_1 and b_2 are two different bins, $|b_1|$ and $|b_2|$ is the number of tokens in b_1 and b_2 respectively, $c(v)$ and $c(w)$ are the number of words of token v and token w. which are in b_1 and b_2 separately. The idea of this cost is to compute the probability that a word from bin b_1 is not equal to a word from bin b_2.

$$cost(b_1, b_2) = Prob(v \neq w, v \in b_1, w \in b_2)$$

The final confusion network is rescored with a 5-gram language model with Kneser-Ney smoothing. To generate the final output, we need to find the best (minimum-cost) path through the rescored confusion network. In the best path every bin in the network contributes only one word to the output.

Ordering the systems for incremental combination and use of different language models were the two components of the pipeline that were experimented with for WMT'2010 shared task. The following sections describe these variations in detail.

3 Learning to Order Systems for Combination

Determining the order in which systems are aligned is critical step in our system combination process. The first few aligned translations/systems determine the word ordering in the final output and have a significant influence on the final translation quality. For the baseline combination the systems are aligned in the increasing order of (TER-BLEU) scores. TER and BLEU (Papineni et.al, 2002) scores are calculated over all the sentences in the training set. This approach to ordering of systems is static and results in a global order for all the source segments. An alternative approach is to learn local order of systems for every source sentence using a SVM ranker.

3.1 SVM Rank Method

This section describes an approach to order systems for alignment using SVMs (Karakos et.al, 2010). For each system output a number of features are generated, the features fall broadly under the following three categories:

N-gram Agreements

These features capture the percentage of hypothesis for a source sentence that contain same n-grams as the candidate translation under consideration. The n-gram matching is position independent because phrases often appear in different orders in sentences with same meaning and correct grammar. The scores for each n-gram are summed and normalized by sentence length. N-grams of length $1 \cdots 5$ are used as five features.

Length Feature

The ratio of length of the translation to the source sentence is a good indication of quality of the translation, for a lengthy source sentence a short translation is most likely to be bad. Here, the ratio of source sentence length to length of the target sentence is calculated.

Language Model Features

Language models for target language are used to calculate perplexity of a given translation. The lower the perplexity the better is the translation quality. We use two different language models: (i) a large static 5-gram language model and (ii)a dynamic language model generated from all the translations of the same source segment. The perplexity values are normalized by sentence length.

Translations in training set are ranked based on (TER-BLEU) scores. An SVM ranker is then trained on this set. The SVM ranker (Joachims, 2002) returns a score for each translation, based on its signed distance from the separating hyperplane. This value is used in the combination process to weight the contribution of systems to the final confusion network scores.

Table 1: Results for all Language pairs on development set

	es-en		fr-en		cz-en		de-en	
Combination	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
BEST SYSTEM	29.27	52.38	26.74	56.88	21.56	58.24	26.53	56.87
BASELINE	28.57	51.61	27.65	55.20	21.01	58.79	26.80	54.54
SVM	28.68	51.99	27.53	55.35	21.56	58.24	26.85	54.9
SVM+NGRAM	29.92	50.92	27.86	55.06	21.80	57.78	27.24	54.86

4 Language Models

In the system combination process, the final confusion networks are rescored with language models. Language models are widely used to ensure a fluent output translation. I explored use of two language models. The first language model was trained on the English side of French-English corpus, UN corpus and English Gigaword corpus made available by WMT. The second language model used counts generated from Google n-grams. It was trained by generating all 1-gram to 5-grams in the system outputs for a source segment and then using the N-gram search engine (Lin et.al, 2010) built over Google n-grams to get the corresponding n-gram counts. The n-gram counts were used to train a 5-gram language model with Kneser-Ney smoothing. SRILM toolkit (Stockle, 2002) was used for training the language models.

The baseline combinations were rescored only with the static language model. I always did a weighted interpolation of the two language models when using n-gram based language model.

5 Results

Results for four language pairs: Spanish-English, French-English, Czech-English and German-English are presented. The training data for WMT'10 was divided into development and test set, consisting of 208 and 247 segments respectively. Table 1 shows TER and BLEU scores on the TEST set for all the four language pairs in the following settings: (i) *Baseline* corresponds to procedure described in section 2, (ii) *SVM* corresponds to using SVM ranker for learning order of systems as described in section 3.1 (iii)*SVM+N-Grams* corresponds to the use of a SVM ranker along with weighted interpolation of n-gram language model and the large static language model. The ranking SVM was trained using SVM-light (Joachims, 2002) with a RBF kernel. Two-fold cross-validation was done to prevent over-fitting on development data. All the scores are with lower-cased outputs, a tri-gram language model was used to true-case the output before the final submission. 1-best output from only the primary systems were used for combination. The number of systems used for combination in each language pair are: 6 for Czech-English, 8 in Spanish-English, 14 in French-English and 16 in German-English. The best results for baseline combination were obtained with 3 systems for Czech-English, 6 systems for German-English, 3 systems for Spanish-English and 9 systems for French-English.

From the results, we conclude that for all language pairs the combinations with SVM and n-gram language models show gain over all the other settings in both TER and BLEU evaluations. However, use of SVM with only one large language model shows performance degradation on three out of four language pairs. Size of training data (208 segments) could be one reason for the degradation and this issue needs further investigation. For the final submission, the settings that performed the best on $\frac{(TER-BLEU)}{2}$ scale were chosen.

6 Conclusion

The system combination task gave us an opportunity to evaluate enhancements added to the JHU system combination pipeline. Experimental results show that web-scale language models can be used to improve translation quality, this further underlines the usefulness of web-scale resources like Google n-grams. Further investigation is needed to completely understand the reasons for inconsistency in the magnitude of gain across different language pairs. Specifically the impact of training data on SVMs for ranking in system combination scenario needs to be analysed.

Acknowledgments

This work was partially supported by the DARPA GALE program Grant No HR0022-06-2-0001. I would like to thank all the participants of WMT 2010 for their system outputs. I would also like to thank Prof. Damianos Karakos for his guidance and support. Many thanks go to the Center for Language and Speech Processing at Johns Hopkins University for availability of their computer clusters.

References

Almut Silja Hildebrand and Stephan Vogel. 2008. *Combination of Machine Translation Systems via Hypothesis Selection from Combined N-Best Lists.* In MT at work: Proceedings of the Eight Conference of Association of Machine Translation in the Americas, pages 254-261, Waikiki, Hawaii, October. Association for Machine Translations in the Americas.

Almut Silja Hildebrand and Stephan Vogel. 2009. *CMU System Combination for WMT'09*. Proceedings of Fourth Workshop on Statistical Machine Translation,Athen,Greece, March 2009.

Andreas Stockle. 2002. *Srilm - an extensible language modeling toolkit*. In Proceedings International Conference for Spoken Language Processing, Denver, Colarado, September.

Antti-Veikko I. Rosti and Necip Fazil Ayan and Bing Xiang and Spyros Matsoukas and Richard Schwartz and Bonnie J. Dorr 2007. *Combining Outputs from Multiple Machine Translation Systems*. In Proceedings of the Third Workshop on Statistical Machine Transaltion, pages 183-186, Colombus, Ohio, June. Association for Computational Linguistics.

Damianos Karakos and Sanjeev Khudanpur 2008. *Sequential System Combination for Machine Translation of Speech*. In Proceedings of IEEE SLT-08, December 2008.

Damianos Karakos and Jason Smith and Sanjeev Khudanpur 2010. *Hypothesis Ranking and Two-pass Approaches for Machine Translation System Combination*. In Proceedings of ICASSP-2010, Dallas, Texas, March 14-19 2010.

Damianos Karakos and Jason Eisner and Sanjeev Khudanpur and Markus Dreyer. 2008. *Machine Translation system combination using ITG-based alignments*. In Proceedings of ACL-08: HLT, Short Papers, pages 81-84, Colombus, Ohio, June. Association for Computational Linguistics.

Dekang Lin and Kenneth Church and Heng Ji and Satoshi Sekine and David Yarowsky and Shane Bergsma and Kailash Patil and Emily Pitler Rachel Lathbury and Vikram Rao and Kapil Dalwani and Sushant Narsale 2010. *New Tools for Web-Scale N-grams*. In the Proceedings of LREC, 2010.

D. Wu 1997. *Stochastic inversion transduction grammars and bilingual parsing of parallel corpora*. Computational Linguistics, vol.23,no.3,pp.377-403, September 1997.

Kishore Papineni and Salim Roukos and Todd Ward and Wei-Jing Zhu. 2002. *BLEU: A method for automatic evaluation of machine translation*. In Proceedings of 40th Annual Meeting of Association for Computational Linguistics, pages 311-318. Philadelphia, PA, July.

Thorsten Joachims 2002. *Optimizing Search Engines using Clickthrough Data*. In Proceedings of ACM Conference on Knowledge Discovery and Data Mining(KDD), 2002.