System Combination for Machine Translation Using N-Gram Posterior Probabilities

Yong Zhao
Georgia Institute of Technology
Atlanta, Georgia 30332
yongzhao@gatech.edu

Xiaodong He
Microsoft Research
1 Microsoft Way, Redmond, WA 98052
xiaohe@microsoft.com

Abstract

This paper proposes using n-gram posterior probabilities, which are estimated over translation hypotheses from multiple machine translation (MT) systems, to improve the performance of the system combination. Two ways using n-gram posteriors in confusion network decoding are presented. The first way is based on n-gram posterior language model per source sentence, and the second, called n-gram segment voting, is to boost word posterior probabilities with n-gram occurrence frequencies. The two n-gram posterior methods are incorporated in the confusion network as individual features of a log-linear combination model. Experiments on the Chinese-to-English MT task show that both methods yield significant improvements on the translation performance, and an combination of these two features produces the best translation performance.

1 Introduction

In the past several years, the system combination approach, which aims at finding a consensus from a set of alternative hypotheses, has been shown with substantial improvements in various machine translation (MT) tasks. An example of the combination technique was ROVER [1], which was proposed by Fiscus to combine multiple speech recognition outputs. The success of ROVER and other voting techniques give rise to a widely used combination scheme referred as confusion network decoding [2][3][4][5]. Given hypothesis outputs of multiple systems, a confusion network is built by aligning all these hypotheses. The resulting network comprises a sequence of correspondence sets, each of which contains the alternative words that are aligned with each other. To derive the consensus hypothesis from the confusion network, a Viterbi algorithm is typically used by selecting a path with the maximum confidence score among all paths that pass the confusion network [9].

The confidence score of a hypothesis could be assigned in various ways. In [1], voting by frequency of word occurrences is used. In Mangu et al.'s work [6], word posterior probabilities was derived as the sum of the confidences from each system output that contains the word in that position. For machine translation, similar to a phrase-based translation approach, the confidence score is frequently formulated as a log-linear combination of several models. An advantage of the log-linear model is that it allows for an arbitrary integration of additional models. For example, in [4], the confidence score includes word-level posterior probabilities, language model (LM) probabilities, and occurrence counts of several word types. So far, word occurrence information of translation hypotheses were exploited, yet sequential information of the hypotheses has not been taken advantage of.

In this paper, we propose using n-gram posterior probabilities that are estimated over multiple translation hypotheses in the hope of improving the consensus translation quality. Intuitively, we will prefer the consensus translation that has high n-gram agreement with the system

outputs. The idea is similar to n-gram posterior probability introduced in [7]. Here, we generalize it in the context of the multiple system combination, where n-grams events of the confusion network may be unseen, and scores of translation systems are not directly comparable, or even unavailable.

Two ways using n-gram posterior probabilities in the confusion network decoding are investigated. The first way is to build an online language model for each source sentence using the n-best lists of the component systems; the second is to boost word posterior probabilities using n-gram occurrence frequencies. The experimental results are presented in the context of a Chinese-English translation task in past years' NIST Open MT evaluation.

The remaining of the paper is organized as follows: first a brief overview of the confusion network decoding for combining outputs of multiple MT systems is given. Then n-gram posterior probability and its two alternative usages in confusion network decoding are described in Section 3. Section 4 presents the experimental results.

2 System combination for machine translation

One of the most successful approaches for MT system combination is based on confusion network decoding, as described in [4]. A schematic diagram of the framework is illustrated in Figure 1. Given translation hypotheses from multiple MT systems, one of the hypotheses is first selected as the backbone for the use of hypothesis alignment. This is usually done by a sentence-level minimum Bayes risk (MBR) method, which selects a hypothesis that has the minimum average distance to all the hypotheses. The confusion network is constructed by aligning all these hypotheses against the backbone. Words that align to each other are grouped into a correspondence set, constituting competition links of the confusion network. Each path in the network passes exactly one link from each correspondence set. The final consensus output relies on a Viterbi decoding procedure, which chooses a path with the maximum confidence score among all paths that pass the confusion network.

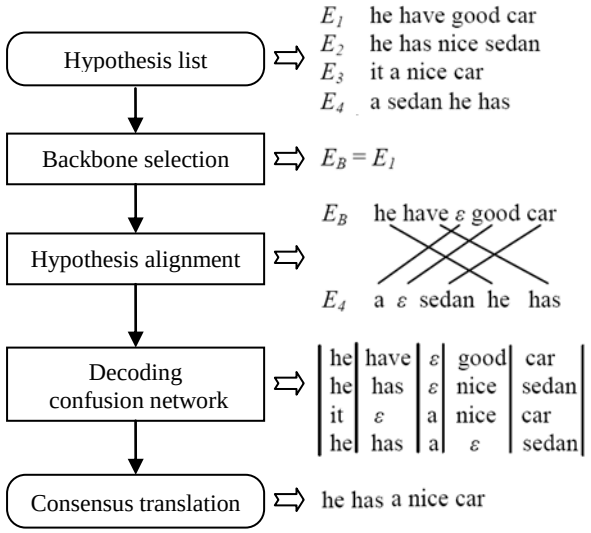

Figure 1: Confusion-network-based MT system combination.

The confidence score of hypotheses is usually formalized as a log-linear sum of several feature functions. Given a source language sentence F, the total confidence of a target language hypothesis $E = (e_1, \dots, e_L)$ in the confusion network is represented as:

$$logP(E|F) = \sum_{l=1}^{L} log\, P(e_l|l, F) + \lambda_1 logP_{LM}(E) + \lambda_2 N_{nulls}(E) + \lambda_3 N_{words}(E) \qquad (1)$$

Where the included feature functions are word posterior probability $P(e_l|l,F)$, language model probability $P_{LM}(E)$, the number of null words N_{nulls} in E, and the number of words N_{words} in E. An advantage of the log-linear model is that it allows for the incorporation of arbitrary feature functions. The model parameter λ_i reflects the contribution of the corresponding feature to the final confidence score. In general, the model parameters λ_i can be obtained by directly optimizing a translation evaluation metric, say BLEU, on a held-out set.

3 N-gram posterior probabilities

The n-gram posterior probability is estimated over all translation hypotheses for a given source sentence. It is a generalization of word posterior probabilities; however, different from word posterior probabilities, n-gram posterior probabilities take advantage of the word sequential information of the system outputs. Intuitively, we would prefer a consensus output that has high n-gram agreement with the system outputs.

It was also argued that the confusion network decoding may introduce an undesirable insertion of words to break coherent phrases generated by the individual systems [8]. Breaking coherent phrases leads to more degradation in terms of BLEU score than the use of TER. The introduction of n-gram posterior probabilities may protect the system combination from this type of mistakes.

Given an input sentence F, the fractional count $C(e_1^n|F)$ of an n-gram e_1^n is defined as:

$$C(e_1^n|F) = \sum_{E' \in \boldsymbol{E}^h} \sum_{l=n}^{L} P(E'|F)\, \delta(e'^{l}_{l-n+1}, e_1^n) \tag{2}$$

where $\delta(\cdot,\cdot)$ denotes the Kronecker function, and $P(E'|F)$ is the posterior probability of translation hypothesis E', which is expressed as the weighted sum of the system specific posterior probabilities through the systems that contains hypothesis E',

$$P(E|F) = \sum_{k=1}^{K} w_k P(E|S_k, F) 1(E \in \boldsymbol{E}_{S_k}) \tag{3}$$

where w_k is the weight for the posterior probability of the k^{th} system S_k, and $1(\cdot)$ is the indicator function.

One way to obtain the system specific posteriors is to derive it as a function of the confidence scores outputted by these individual systems; However, in most cases, scores of these translation systems are not directly comparable, or even unavailable. Experimentally, it turns out that a simple rank-based scoring scheme would work effectively. If translation hypothesis E_r is the r^{th} best output in the n-best list of system S_k, posterior $P(E_r|S_k,F)$ is approximated as:

$$P(E_r|S_k,F) = \frac{1/(1+r)^{\eta}}{\sum_{r'=1}^{||\boldsymbol{E}_{S_k}||} 1/(1+r')^{\eta}} \tag{4}$$

where η is a rank smoothing parameter.

The remaining of this section presents two ways of applying n-gram posterior probabilities in the confusion network decoding.

3.1 N-gram posterior language model

A straightforward approach using n-gram posterior probabilities is to formulate it as a language model (LM). In this way, we build an n-gram posterior language model for each source sentence using the n-best lists outputted by the component systems. Similar to a regular LM, the n-gram posterior LM will be added as an additional feature in the log-linear model for confusion network decoding. The n-gram posterior LM is:

$$P(e_l|e_{l-n+1}^{l-1},F)=\frac{C(e_{l-n+1}^{l}|F)}{C(e_{l-n+1}^{l-1}|F)} \tag{5}$$

The n-gram posterior probability of the entire sequence of hypothesis E is obtained as:

$$P_{LM}(E|F)=\prod_{l=n}^{L}P\left(e_l|e_{l-n+1}^{l-1},F\right) \tag{6}$$

It is necessary to smooth the n-gram probabilities, since we may meet unseen n-grams in the confusion network. Here we choose linear interpolation to combine n-grams of different orders.

$$P_{smoot\ h}\left(e_l|e_{l-n+1}^{l-1},F\right)=\sum_{m=1}^{n}\alpha_m P(e_l|e_{l-m+1}^{l-1},F) \tag{7}$$

To learn the interpolation weights, a widely used method is to minimize the perplexity of the language model on a held-out data set. Yet, for the simplicity, we will integrate the learning of the interpolation weights with the log-linear model training. Experiments show that reliable results can be obtained by this end-to-end approach.

3.2 N-gram segment voting

The second usage of the n-gram posterior probability is to boost the word posterior probability using n-gram occurrence frequencies, a method referred as n-gram segment voting (NGV).

The n-gram segment voting is derived as follows. If we choose words as the basic type of segments, the risk of a hypothesis becomes:

$$R(E)=\sum_{E^{'}\in\boldsymbol{E}^{h}}P(E^{'}|F)\sum_{l=1}^{L}L(e_l^{'},e_l)=\sum_{l=1}^{L}\sum_{e_l^{'}\in e_l^{h}}L(e_l^{'},e_l)P(e_l^{'}|l,F) \tag{8}$$

where the local risk of word e_l at position l is:

$$R_l(e_l)=\sum_{e_l^{'}\in e_l^{h}}L(e_l^{'},e_l)P(e_l^{'}|l,F) \tag{9}$$

Note that the local risk under a 0-1 loss function reduces to the expression of word-level posterior probabilities.

The basic segments may not be confined to word level. Considering n-gram tokens as basic segments, the local risk of n-gram segment e_{l-n+1}^{l} at position l is:

$$R_l\left(e_{l-n+1}^{l}\right)=\sum_{m=1}^{n}\sum_{e^{'}{}_{l-m+1}^{l}}L\left(e^{'}{}_{l-m+1}^{l},e_{l-m+1}^{l}\right)P(e^{'}{}_{l-m+1}^{l}|l,F) \tag{10}$$

where $P(e_{l-n+1}^{l}|l,F)$ standards for posterior probability of n-gram e_{l-n+1}^{l} that occurs at position $(l-n+1),\dots,l$.

If applying a 0-1 loss function to Equation (10), the n-gram-level local risk becomes a sum of the n-gram posteriors,

$$P_{NGV}\left(e_{l-n+1}^{l}|l,F\right)=\sum_{m=1}^{n}P(e_{l-m+1}^{l}|l,F) \tag{11}$$

But this kind of n-gram occurrence posterior encounters a severe sparse-data problem. When n is larger than 1, it is difficult to estimate n-gram posterior at a specific position. Hence, for n larger than 1, we approximate probability $P(e_{l-m+1}^{l}|l,F)$ of n-gram e_{l-n+1}^{n} with its occurrence posterior regardless of its position,

$$P_{NGV}\left(e_{l-n+1}^{l}|l,F\right)=P(e_{l}|l,F)+\sum_{m=2}^{n}\beta_{m}C(e_{l-m+1}^{l}|\,F) \tag{12}$$

A direct interpretation of n-gram segment voting is that we boost the posterior probability of words with n-gram occurrence frequencies.

4 Experimental results

In this section, we evaluate the proposed n-gram posterior probabilities on the Chinese-to-English (C2E) test in the past NIST Open MT Evaluations. The experimental results are reported using the case insensitive BLEU score as evaluation metric [10].

The development set, which is used for system combination parameter training, contains 1002 sentences that are sampled from the previous NIST MT Chinese-to-English test sets: 35% from MT04, 55% from MT05, and 10% from MT06-newswire. The test set is the left part of NIST Chinese-to-English test sets of MT04-MT05, MT06-newswire and MT06-newsgroup. Both development and test sets have four reference translations per sentence.

Outputs from a total of eight single MT systems were combined for consensus translations. These selected systems are based on various translation paradigms, such as phrasal, hierarchical, and syntax-based systems. Each system produces 10-best hypotheses per translation, and thus we have at most 80 hypotheses per sentence for system combination. In the experiments, the BLEU score range for the eight individual systems are from xx to xx on the dev set and from xx to xx on the test set.

Table 1 shows the experimental results using n-gram posterior probabilities to build an online language model for each source sentence in confusion network decoding. The baseline system combination is the one proposed in [5]. Using 2-gram and 3-gram language models yields BLEU of 41.74%, which outperforms the baseline system by 0.49 BLUE points. The use of 1-gram posterior language model, which amounts to word occurrence probabilities of the system outputs, does not produce improvements. Also, it is observed that the performance on the test set degrades, when we increase the order of n-gram to 4.

Table 1: Evaluation results using the n-gram posterior language model.

	Dev BLEU %	Test BLEU %
Best single system	37.80	37.02
Baseline	42.67	41.25
1-gram	42.67	41.21
2-gram	43.24	**41.74**
3-gram	43.27	**41.74**
4-gram	43.30	41.64

Table 2 shows the performance using n-gram segment voting. The use of 4-gram yields the best BLUE of 41.58%, which is different from the case of n-gram posterior language model. We explain that though the estimate of 4-gram probabilities is still inaccurate, the modificative

nature of the n-gram segment voting method, which adds the n-gram occurrence frequency to the word posterior probability, makes it less sensitive to zero n-gram probabilities.

Table 2: Evaluation results of MT system combination using n-gram segment voting.

	Dev BLEU %	Test BLEU %
Best single system	37.80	37.02
Baseline	42.67	41.25
1-gram	42.67	41.25
2-gram	42.72	41.46
3-gram	42.85	41.38
4-gram	43.10	**41.58**

The improvements of both n-gram posterior methods lead us to explore the combination of these two methods, though they bear the same type of information. The confusion network decoding using n-gram posterior LM and n-gram segment voting of orders of interest is reported in Table 3. The configure using 3-gram posterior LM and 4-gram segment voting, which are the best orders for each method when evaluated individually, yields the best BLEU score of 41.89%, an absolute improvement of 0.64 BLEU point over the baseline.

Table 3: Evaluation results using the combination of n-gram posterior language model and n-gram segment voting.

	Dev BLEU %	Test BLEU %
Best single system	37.80	37.02
Baseline	42.67	41.25
1-gram LM + 1-gram voting	42.69	41.21
2-gram LM + 2-gram voting	42.88	41.41
3-gram LM + 3-gram voting	43.24	41.80
4-gram LM + 4-gram voting	43.26	41.63
3-gram LM + 4-gram voting	43.38	**41.89**

5 Conclusion

In this paper, we explored n-gram posterior probabilities to take advantage of word sequential information of translation outputs of multiple MT systems. Two ways using n-gram posterior probabilities in the framework of confusion network decoding were presented. The first is to produce an online language model for each source sentence using n-gram posterior probabilities, and the second is to boost the word posterior probability with the n-gram occurrence frequencies. Our experiments on the Chinese-English NIST task showed that both methods using n-gram posterior probabilities yield significant improvements on the translation quality. Moreover, the combination of these two methods produced the best translation performance with BLEU score of 41.89%.

There are several ways to improve the proposed approach. For example, by cumulating n-gram posterior probabilities over system outputs for the preceding source sentences, we may create a cache n-gram posterior language model. The cache language model may facilitate an accurate estimation of n-gram posteriors.

A general heuristic of our work on n-gram posterior probabilities is that we may improve system combination performance by building models to capture patterns, no matter short or long span, which are common among systems hypotheses.

References

[1] J.G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)," in *Proc. IEEE ASRU Workshop*, 1997.

[2] S. Bangalore, G. Bordel, and G. Riccardi, "Computing consensus translation from multiple machine translation systems," in *Proc. ASRU*, 2001.

[3] E. Matusov, N. Ueffing, and H. Ney, "Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment," in *Proc. EACL*, 2006.

[4] A.-V.I. Rosti, S. Matsoukas, and R. Schwartz, "Improved Word-Level System Combination for Machine Translation." In *Proc. ACL*, 2007.

[5] X. He, M. Yang, J. Gao, P. Nguyen, and R. Moore, "Indirect-HMM-based hypothesis alignment for combining outputs from machine translation systems," in *Proc. EMNLP*, 2008.

[6] L. Mangu, E. Brill, and A. Stolcke, "Finding Consensus in Speech Recognition: Word Error Minimization and Other Applications of Confusion Networks, " *Computer Speech and Language*, 14(4):373-400, 2000.

[7] R. Zens and H. Ney, "N-Gram posterior probabilities for statistical machine translation," in *Proc. HLT-NAACL*, 2004.

[8] K.C. Sim, W.J. Byrne, M.J.F. Gales, H. Sahbi and P.C. Woodland, "Consensus network decoding for statistical machine translation system combination." in *Proc. ICASSP*, 2007.

[9] V. Goel, S. Kumar, and W. Byrne, "Segmental minimum Bayes-risk decoding for automatic speech recognition." *IEEE transactions on Speech and Audio Processing*, vol. 12, no. 3, 2004.

[10] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "BLEU: a method for automatic evaluation of machine translation." in *Proc. ACL*, 2002.