Generalized Minimum Bayes Risk System Combination

Kevin Duh, Katsuhito Sudoh, Xianchao Wu, Hajime Tsukada, Masaaki Nagata
NTT Communication Science Laboratories
2-4 Hikari-dai, Seika-cho, Kyoto 619-0237, JAPAN
kevin.duh@lab.ntt.co.jp

Abstract

Minimum Bayes Risk (MBR) has been used as a decision rule for both single-system decoding and system combination in machine translation. For system combination, we argue that common MBR implementations are actually not correct, since probabilities in the hypothesis space cannot be reliably estimated. These implementations achieve the effect of consensus decoding (which may be beneficial in its own right), but does not reduce Bayes Risk in the true Bayesian sense.

We introduce *Generalized* MBR, which parameterizes the loss function in MBR and allows it to be optimized in the given hypothesis space of multiple systems. This extension better approximates the true Bayes Risk decision rule and empirically improves over MBR, even in cases where the combined systems are of mixed quality.

1 Introduction

Minimum Bayes Risk (MBR) is a theoretically-elegant decision rule that has been used for single-system decoding and system combination in machine translation (MT). MBR arose in Bayes decision theory (Duda et al., 2000) and has since been applied to speech recognition (Goel and Byrne, 2000) and machine translation (Kumar and Byrne, 2004). The idea is to choose hypotheses that minimize *Bayes Risk* as oppose to those that maximize posterior probability. This enables the use of task-specific loss functions (e.g BLEU).

However, the definition of Bayes Risk depends critically on the posterior probability of hypotheses. In single-system decoding, one could approximate this probability using model scores. However, for system combination, the various systems have incompatible scores. In practice, MT designers resort to uniform probability, but the result is that the chosen hypothesis no longer has anything to do with Bayes Risk. This hypothesis can be seen as a *consensus* of multiple hypotheses, and in practice the consensus translation is often good, but it cannot be accurately thought of as MBR.

Here, we propose a method that better achieves MBR in system combination settings. The insight is to generalize the loss function in the MBR equation and allow it to be parameterized. The parameters are then tuned on a small development data so that the loss function is converted to one that gives low Bayes Risk under the assumption of uniform posteriors. We will show that a small bitext is sufficient for tuning this generalized loss, and that it vastly outperforms the conventional MBR approach in system combination.

In the following, we first review MBR and explain the difficulty in applying it to system combination (Section 2). Then, we propose our Generalized MBR (Section 3) and evaluate it under the NTCIR Patent Translation tasks (Section 4). Finally, we conclude in Section 5.

2 The Difficulty with MBR

Consider the task of translation from a French sentence (f) to an English sentence (e). Our goal is to find a decision rule $\delta(f) \rightarrow e'$, which takes f as input and generates a e' as output, to minimize the expected loss (i.e. Bayes Risk) over the possible space of sentence pairs ($p(e, f)$):

$$E_{p(e,f)}[L(\delta(f)|e)] \quad (1)$$

Note that we write loss $L(\delta(f)|e)$ rather than the conventional $L(\delta(f), e)$ to emphasize that it is asymmetric. The loss allows us to incorporate task-specific knowledge. For example, with 1-BLEU as the loss function, we can quantify that the sentence with 2-word mismatch is preferable

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 1356–1360,
Chiang Mai, Thailand, November 8 – 13, 2011.

to one that has 3-word mismatch, even though both do not perfectly match the reference. The incorporation of the task-specific loss is why MBR is attractive in applications.

What decision rule minimizes the expected loss? By reorganizing Eq. 1 as follows:

$$\begin{aligned} E_{p(e,f)}[L(\delta(f)|e)] &= \sum_{e,f} L(\delta(f)|e)p(e,f) \\ &= \sum_{e,f} L(\delta(f)|e)p(e|f)p(f) \\ &= \sum_{f} \left[\sum_{e} L(\delta(f)|e)p(e|f) \right] p(f) \qquad (2) \end{aligned}$$

we observe that expected loss can be minimized if the term in the bracket (known as the *conditional risk*) is minimized for each f:

$$\begin{aligned} & \arg\min_{\delta(f)} \sum_{e} L(\delta(f)|e)p(e|f) \qquad (3) \\ \approx \; & \arg\min_{e' \in N(f)} \sum_{e \in N(f)} L(e'|e)p(e|f) \qquad (4) \end{aligned}$$

Eq. 3 is the Minimum Bayes Risk (MBR) decision rule. Eq. 4 is the N-best approximation commonly used in practice: $N(f)$ contains the set of hypotheses in the N-best list, and the argmin and sum is only performed within this finite set. There are two difficulties with Eq. 4:

1. The N-best approximation is much smaller than the *true* space of *all* English hypotheses in the argmin and sum of Eq. 3. The approximation in the argmin causes search errors, while the approximation in the sum introduces bias. This problem can be somewhat mitigated by increasing the N-best list size or extending this space using lattices and hypergraphs (Tromble et al., 2008; DeNero et al., 2009; Kumar et al., 2009). We do not address this issue here.

2. The posterior probability $p(e|f)$ in Eq.3 and Eq. 4 refers to the *true* posterior probability arising from $E_{p(e,f)}[\cdot]$ in the derivation of Eq. 2. In practice, this can only be estimated from the MT decoder's model scores:

$$p(e|f) \approx \frac{(\exp \sum_i \lambda_i h_i(e,f))^\alpha}{\sum_{e' \in N(f)} (\exp \sum_i \lambda_i h_i(e',f))^\alpha} \qquad (5)$$

where $h_i(e,f)$ are features, λ_i are feature weights, and α is a scaling factor that determines the flatness of the posterior distribution (Ehling et al., 2007). It is important to emphasize that we are *assuming* that the decoder's score is an accurate surrogate for the true posterior distribution $p(e|f)$.

The second difficulty poses a particular problem for system combination. Although the assumption in Eq. 5 is reasonable for single-system MT, it becomes unclear how to compare the model scores $\sum_i \lambda_i h_i(e,f)$ in a multi-system setting. To illustrate, consider two MT systems, their 2-best lists, and corresponding model scores:

- System A: e_1, score=7; e_2, score=3;
- System B: e_3, score=90; e_4, score=10;

It is unclear what is the ranking of posterior probabilities in the space of these four hypotheses. The possibilities include:

- $p(e_1|f) > p(e_2|f) > p(e_3|f) > p(e_4|f)$,
- $p(e_3|f) > p(e_4|f) > p(e_1|f) > p(e_2|f)$,
- $p(e_1|f) > p(e_3|f) > p(e_2|f) > p(e_4|f), \ldots$

From the model scores, we can assume that $p(e_1|f) > p(e_2|f)$ and $p(e_3|f) > p(e_4|f)$ but we cannot say anything about how, e.g., $p(e_1|f)$ and $p(e_3|f)$ compare because the scores are not calibrated across systems. If we cannot even rank the posteriors, there is little hope of estimating its actual values.

Due to this difficulty, previous work in MBR system combination disregard the estimation and assume that $p(e|f)$ is an uniform distribution. The effect is *consensus decoding*, i.e. picking a sentence most similar to others in the N-best list. Consensus decoding may be beneficial in its own right, as shown by positive results in (de Gispert et al., 2009; Sim et al., 2007), but the consensus rule and the MBR rule are *different*.

In fact, the consensus rule may suffer if the N-best list contains many poor translations that are similar to each other. On the other hand, if these poor translations all have small posterior (which ought to be), it does not affect the MBR rule whatsoever. Unfortunately, the bottleneck is the difficulty in estimating the posteriors.

3 Generalized MBR

3.1 Theory

The idea of *Generalized* MBR (GMBR) is to parameterize the loss function in Eq. 4 and allow it to adapt to the hypothesis space of a set of given systems. Specifically, we write a loss function $L(e'|e;\boldsymbol{\theta})$, parameterized by $\boldsymbol{\theta}$, as a linear combination of sub-components:

$$L(e'|e;\theta) = \sum_{k=1}^{K} \theta_k L_k(e'|e) \tag{6}$$

The sub-components are related to the original loss function in some fashion. For example, suppose the desired loss function is 1-BLEU. Then the sub-components could be: $L_1(e'|e)$=1 - 1gramPrecision, $L_2(e'|e)$=1 - 2gramPrecision, $L_3(e'|e)$=1 - 3gramPrecision, $L_4(e'|e)$=1 - 4gramPrecision, $L_5(e'|e)$=brevaltyPenalty.

The GMBR rule is defined generically as:

$$\arg\min_{e' \in N(f)} \sum_{e \in N(f)} L(e'|e;\boldsymbol{\theta}) p(e|f) \tag{7}$$

And in the case of system combination, we will assume uniform $p(e|f)$ and re-write the GMBR decision rule as:

$$\arg\min_{e' \in N(f)} \sum_{e \in N(f)} L(e'|e;\boldsymbol{\theta}) \frac{1}{|N(f)|} \tag{8}$$

$$= \arg\min_{e' \in N(f)} \sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e'|e) \tag{9}$$

Our goal is to minimize the expected loss (Eq. 1) under the constraint of uniform $p(e|f)$. The central idea is this: we will tune $\theta_k, k = 1, \ldots, K$ so that the generalized loss in the uniform hypotheses space gives the same decision as the original loss in the true space $p(e|f)$.

This can be done if a small dev set is available: For any two hypotheses e_1, e_2, and a reference e_r (not in $N(f)$) we first compute the true loss: $L(e_1|e_r)$ and $L(e_2|e_r)$. If $L(e_1|e_r) < L(e_2|e_r)$, then we would want $\boldsymbol{\theta}$ such that:

$$\sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_1|e) < \sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_2|e) \tag{10}$$

so that GMBR would select the hypothesis achieving lower loss. Conversely, if e_2 is a better hypothesis, then we want the opposite relation: $\sum_e \sum_k \theta_k L_k(e_1|e) > \sum_e \sum_k \theta_k L_k(e_2|e)$. Thus, in light of the fact that posterior probabilities $p(e|f)$ are not reliable, we directly compute the true loss (using a development set) and ensure that our GMBR decision rule minimizes this loss.

The disadvantage of GMBR is, of course, that a development set is needed. Note, however, that MBR may also require tuning the global scaling factor (Eq. 5). Empirically, we observe that a small set (500 sentences) seems sufficient.

Algorithm 1 Tuning $\boldsymbol{\theta}$ for GMBR

Input: Development data $\mathcal{D}$, with $(e_r, f) \in \mathcal{D}$
Input: N-best list $N(f)\ \forall f \in \mathcal{D}$.
Input: Regularization hyperparameter c
Output: $\theta_k, k = 1, \ldots, K$ such that Eq. 9 minimizes $L()$ on $\mathcal{D}$.

1: $\mathcal{P} = \emptyset$
2: **for** $f \in \mathcal{D}$ **do**
3: Compute true loss $L(e|e_r)\ \forall e \in N(f)$
4: **for** All pair $e_i, e_j \in N(f)$ **do**
5: Add (e_i, e_j) to $\mathcal{P}$ if $L(e_i|e_r) < L(e_j|e_r)$
6: **end for**
7: **end for**
8: $\arg\min_{\boldsymbol{\theta}} ||\theta||^2 + c \sum_{ij} \xi_{ij}$
s.t. $\sum_k \theta_k C_k(e_j) - \sum_k \theta_k C_k(e_i) \geq 1 - \xi_{ij}$
$\forall (e_i, e_j) \in \mathcal{P}$

3.2 Implementation

We now describe how GMBR and the tuning procedure can be implemented in practice. First, note that we can reorganize the sums in the GMBR decision rule: $\sum_e \sum_k \theta_k L_k(e'|e) = \sum_k \theta_k \sum_e L_k(e'|e) = \sum_{k=1}^{K} \theta_k C_k(e')$, where $C_k(e') = \sum_e L_k(e'|e)$ represents the combined loss for e'. So we first compute $C_k(\cdot)$ for all hypotheses, for an $O(|N(f)|^2)$ run-time. To find the GMBR decision then requires a search $\arg\min_{e' \in N(f)} \sum_{k=1}^{K} \theta_k C_k(e')$. So in test, GMBR is on the same order as conventional MBR.

To tune $\boldsymbol{\theta}$, we first extract all pairs of hypotheses where a difference exists in the true loss, then optimize $\boldsymbol{\theta}$ in a formulation similar to RankSVM (Joachims, 2006). The pair-wise nature of Eq. 10 makes the problem amenable to solutions in "learning to rank" literature (He et al., 2008a). The pseudocode is shown in Algorithm 1. The RankSVM (line 8) tries to satisfy the relations (Eq. 10) in its constraints while allowing for some slack ξ, whose amount depends on hyperparameter c.

4 Experiments

We experiment with the NTCIR-9 (2011) English-to-Japanese Patent Translation task[1]. This includes 3 million sentences for training individual MT systems; the official dev set is split into 1000 sentences for MERT of individual systems, 500 for system combination optimization (MBR, GMBR), and 500 for final evaluation. We combine three systems:

- Phrase-based Moses with lexical reordering, distortion=6 (Koehn and others, 2007)
- Forest-to-string system (Mi et al., 2008)
- Weighted finite-state Transducer (WFST) (Zhou et al., 2006) with rule-based reordering as preprocessing (Isozaki et al., 2010b).

Each system generates a 100-best list, so our system combination task involves hypothesis selection out of 300 hypotheses. As evaluation measure, we focus on BLEU, Normalized Kendall's Tau (NKT), a metric that has been shown to correlate well with humans on this language pair (Isozaki et al., 2010a)[2], and a combination thereof. The loss function used for MBR is therefore the sum of BLEU and NKT. For GMBR, the sub-components of this loss function are derived from the n-gram precisions, brevity penalty, and Kendall's score. We also multiply the n-gram precisions with the Kendall score as additional loss sub-components. Finally we add identity features indicating which of the three systems the hypothesis comes from, for a total of $K = 14$ sub-components. The hyperparameter c in Algorithm 1 is chosen by 80/20% cross-validation from the set $\{0.001, 0.01, 0.1, 1, 10, 100\}$.

The test scores for MBR, GMBR, and the single systems are shown in Table 1. The single systems are anonymized by A, B, C and sorted by decreasing performance. The top1 indicates the first hypothesis in the 100-best list, while bottom1 indicates the 100st (last) hypothesis. Observations:

1. GMBR outperforms MBR on all metrics.

2. GMBR is able to improve upon the best single system (A), despite the fact that a poor system (C) is included. This implies that criteria like Eq. 10 is effective.

3. For MBR, the inclusion of C drastically degrades performance since it implements consensus decoding, not Bayes Risk.

We also summarize results for Chinese-English and Japanese-English tasks in NTCIR9. The system combination setting is similar (3-way combination of 100-best lists) but uses different MT systems. In Chinese-English, GMBR outperforms the best single system by 1 BLEU point (32.08 vs. 31.08); in Japanese-English, GMBR outperforms by 1.85 BLEU points (29.39 vs. 27.54).

We conclude that GMBR is a robust method for system combination. It consistently improves over the top system, even when the combinations are of varying quality (e.g., the range of BLEU score in the 300-best list can be more than 10 BLEU points between A-top1 and C-bottom1). This degrades MBR and consensus decoding, but does not impact GMBR because these poor translation would achieve high loss on the development set, and therefore $\boldsymbol{\theta}$ will be optimized away from them.[3]

	BLEU	NKT	(BLEU+NKT)/2
GMBR	**36.65**	**77.50**	**57.08**
MBR	35.45	76.25	55.85
A top1	35.87	76.87	56.37
B top1	34.20	75.93	55.07
C top1	24.23	67.68	45.96
A bottom1	34.92	76.20	55.56
B bottom1	33.97	75.99	54.93
C bottom1	22.95	65.99	44.47
oracle	45.82	84.32	65.07

Table 1: Test results on English-Japanese.

5 Conclusions

We introduced *Generalized* MBR, which enables one to adapt the loss function of MBR to a given hypothesis space. By tuning this generalized loss under the constraint of uniform posteriors, we show that GMBR can consistently outperform MBR in system combination. Future work includes (1) combination with methods that can generate novel hypotheses (Rosti et al., 2007; He et al., 2008b; Matusov et al., 2006; Bangalore et al., 2001), and (2) comparison with recent work that attempts to directly estimate posteriors with mixture models (Duan et al., 2010).

[1] `http://ntcir.nii.ac.jp/PatentMT/`

[2] Code available at `http://www.kecl.ntt.co.jp/icl/lirg/ribes`

[3] It's worth noting that system identity features account for less than 30% of weights in all GMBR systems, implying that the flexibility of adjustable loss function is important and a straightforward weighted version of MBR is insufficient.

References

Srinivas Bangalore, German Bordel, and Giuseppe Riccardi. 2001. Computing consensus translation from multiple machine translation systems. In *ASRU*.

Adria de Gispert, Sami Virpioja, Mikko Kurimo, and Willliam Byrne. 2009. MBR combination of translation hypotheses from alternative morphological decompositions. In *NAACL*.

John DeNero, David Chang, and Kevin Knight. 2009. Fast consensus decoding over translation forests. In *ACL*.

Nan Duan, Mu Li, Dongdong Zhang, and Ming Zhou. 2010. Mixture model-based minimum bayes risk decoding using multiple machine translation systems. In *COLING*.

Richard Duda, Peter Hart, and David Stork. 2000. *Pattern Classification*. Wiley-Interscience, 2nd edition.

Nicola Ehling, Richard Zens, and Hermann Ney. 2007. Minimum bayes risk decoding for BLEU. In *ACL Demo and Poster session*.

Vaibhava Goel and William Byrne. 2000. Minimum bayes-risk automatic speech recognition. *Computer Speech and Language*, 14(2):115–135.

Chuan He, Cong Wang, Yi xin Zhong, and Rui fan Li. 2008a. A survey on learning to rank. In *International Conference on Machine Learning and Cybernetics*.

Xiaodong He, Mei Yang, Jianfeng Gao, Patrick Nguyen, and Robert Moore. 2008b. Indirect-HMM-based hypothesis alignment for combining outputs from machine translation systems. In *EMNLP*.

H. Isozaki, T. Hirao, K. Duh, K. Sudoh, and H. Tsukada. 2010a. Automatic evaluation of translation quality for distant language pairs. In *EMNLP*.

Hideki Isozaki, Hajime Tsukada, Katsuhito Sudoh, and Kevin Duh. 2010b. Head finalization: a simple reodering rule for SOV langauges. In *ACL Workshop on Statistical Machine Translation (WMT)*.

T. Joachims. 2006. Training linear SVMs in linear time. In *KDD*.

P. Koehn et al. 2007. Moses: open source toolkit for statistical machine translation. In *ACL*.

Shankar Kumar and William Byrne. 2004. Minimum bayes-risk decoding for statistical machine translation. In *HLT-NAACL*.

Shankar Kumar, Wolfgang Macherey, Chris Dyer, and Franz Och. 2009. Efficient minimum error rate training and minimum bayes risk decoding for translation hypergraphs and lattices. In *ACL*.

Evgeny Matusov, Nicola Ueffing, and Hermann Ney. 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. In *EACL*.

Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-based translation. In *ACL*.

Antti-Veikko I. Rosti, Necip Fazil Ayan, Bing Xiang, Spyridon Matsoukas, Richard M. Schwartz, and Bonnie J. Dorr. 2007. Combining outputs from multiple machine translation systems. In *NAACL-HLT*.

Khe Chai Sim, William J. Byrne, Mark J.F. Gales, Hichem Sahbi, and Phil C. Woodland. 2007. Consensus network decoding for statistical machine translation system combination. In *ICASSP*.

Roy Tromble, Shankar Kumar, Franz Och, and Wolfgang Macherey. 2008. Lattice minimum bayes-risk decoding for statistical machine translation. In *EMNLP*.

Bowen Zhou, Stanley Chen, and Yuqing Gao. 2006. Folsom: A fast and memory-efficient phrase-based approach to statistical machine translation. In *IEEE Spoken Language Technology Workshop*.