Listwise Ranking Functions for Statistical Machine Translation

Meng Zhang, Yang Liu, Huanbo Luan, Maosong Sun

***Abstract*—Decision rules play an important role in the tuning and decoding steps of statistical machine translation. The traditional decision rule selects the candidate with the greatest potential from a candidate space by examining each candidate individually. However, viewing each candidate as independent imposes a serious limitation on the translation task. We instead view the problem from a ranking perspective that naturally allows the consideration of an entire list of candidates as a whole through the adoption of a listwise ranking function. Our shift from a pointwise to a listwise perspective proves to be a simple yet powerful extension to current modeling that allows arbitrary pairwise functions to be incorporated as features, whose weights can be estimated jointly with traditional ones. We further demonstrate that our formulation encompasses the minimum Bayes risk (MBR) approach, another decision rule that considers restricted listwise information, as a special case. Experiments show that our approach consistently outperforms the baseline and MBR methods across the considered test sets.**

***Index Terms*—Statistical machine translation, listwise ranking function, discriminative reranking.**

I. Introduction

MACHINE translation (MT) strives to produce a target translation from a given source sentence. In statistical machine translation (SMT), this process is generally modeled as a probability distribution over the target translations given the source sentence. After decades of development, the probabilistic models used for this purpose have grown from generative models, known as source-channel models [1], to discriminative models, known as maximum entropy or log-linear models [2]. Meanwhile, the typical tuning[1] objectives have evolved from maximum likelihood to task-specific objectives [3].

Over the years, log-linear models with task-specific objectives have gained popularity. Their success hinges on the extensibility of the considered features and the ability to optimize them toward a particular evaluation metric. Despite their popularity, they are subject to several limitations.

This research is supported by the 863 Program (2015AA015407) and the National Natural Science Foundation of China (No. 61522204, 61432013, 61303075). This research is also supported by the Singapore National Research Foundation under its International Research Centre@Singapore Funding Initiative and administered by the IDM Programme. *(Corresponding author: Y. Liu.)*

M. Zhang, Y. Liu, H. Luan, and M. Sun are with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: zmlarry@foxmail.com; liuyang2011@tsinghua.edu.cn; luanhuanbo@gmail.com; sms@tsinghua.edu.cn).

[1]The estimation of the parameters in such a model is traditionally referred to as tuning or development; these terms are analogous to the concept of training in the machine learning literature. We use the term "tuning" throughout this paper, even in machine learning contexts.

One limitation of log-linear models lies in the fact that other candidates are ignored while examining the candidate at hand. More specifically, for N potential translations in an N-best list, log-linear models examine each potential translation individually rather than as a whole, although only the translation with the greatest potential in the N-best list is relevant for both tuning and decoding.

Consider the illustrative example shown in Fig. 1. It consists of a list of four candidate translations. The usual log-linear model assigns each candidate a score that is computed entirely based on that individual translation, referred to as the "pointwise score". This score may not be very accurate, as in this example, where a poor translation receives the highest score and is ranked in the first place in the list. A typical translation system will then output the highest scoring candidate, discarding all other potential translations. However, we observe that the other translations are better in similar ways, while being radically different from the first translation. If we poll the entire list, one of the other three translations might stand out as superior through mutual support. This can be achieved by asking each candidate to report its similarity to all other translations, a process that draws information from the entire list. This similarity is represented by a "listwise score". If we combine both pointwise and listwise information by summing the two types of scores, we find that the second translation becomes the highest scoring one and will be selected.

A number of previous works have partially addressed the limitation that the usual log-linear models face. One line of research has introduced minimum Bayes risk (MBR) decoding [4], which utilizes the MBR decision rule to consider the complete N-best list with the assistance of an evaluation metric in the selection of the translation with the greatest potential. Another line of research has introduced the notion of consensus. This can be seen as a variant of the MBR approach, albeit without theoretical justification, that permits fast decoding [5].

However, none of the above formulations offers the flexibility to incorporate more than one evaluation metric into the model. For example, Kumar and Byrne [4] noted an increase in performance when the metric used for evaluation and decision is "matched". It is possible that other metrics, in combination with the matched metric, could further boost target evaluation performance, but the MBR decision rule forbids such experimentation; this is also true for the consensus approach.

Instead, we view log-linear models from a ranking perspective. From this point of view, log-linear models can be seen as *ranking functions* that select the translation with the greatest potential from an N-best list by choosing the highest

Candidate	Pointwise score	Listwise score	Sum
Sharon with Bush hold talks	1	0.3	1.3
Bush held a talk with Sharon	**0.5**	**1**	**1.5**
Bush holds a talk with Sharon	0.5	0.9	1.4
Bush held talks with Sharon	0.5	0.9	1.4

Figure 1. An illustrative example demonstrating the benefit of incorporating information about the entire candidate list. The original ordering of the list is determined by the pointwise scores, which rank a poor translation in first place. If we additionally compute the listwise scores by placing emphasis on translations that are similar to others, then the sum of the two scores indicates the translation presented in bold as the best.

scoring one (Section III-A). In this regard, the limitation of examining the candidates individually is essentially rooted in the *pointwise* nature of the ranking functions. Naturally, this could be resolved by adopting *listwise ranking functions*. Although general listwise ranking functions are difficult to parametrize, by virtue of the recent work of [6], we are able to represent a listwise ranking function in terms of pairwise functions under a natural assumption (Section III-B). We proceed to demonstrate how these pairwise functions reduce to features in a log-linear model that carry information gathered from the entire N-best list (Section III-C). Unlike in previous works that have also considers the list as a whole, our features can be formed from any pairwise function and any number of pairwise functions. An additional benefit of our feature formulation approach is its simplicity, as these new listwise features can be easily embedded back into the existing log-linear framework and their weights can be adjusted simultaneously with traditional pointwise feature weights to account for the interactions between the two types of features. The joint tuning of pointwise and listwise feature weights is a crucial property for a better-performing system (Section IV-D) and distinguishes our approach from previous work (Section V-B). Furthermore, our formulation naturally encompasses the MBR rule, which, in our framework, would appear as a single feature (Section III-D).

We present experiments conducted on Chinese-English translation and demonstrate promising improvement over the log-linear baseline and the MBR approach (Section IV-C).

II. Background

A. Log-linear Models

Log-linear models are ubiquitous in statistical machine translation for modeling the translation process. They define a probability distribution over a target "English" translation $\mathbf{e}$ given a source "foreign" sentence $\mathbf{f}$ [2]:

$$p\left(\mathbf{e}|\mathbf{f};\boldsymbol{\lambda}\right)=\frac{\exp\left\{\boldsymbol{\lambda}\cdot\boldsymbol{h}\left(\mathbf{e},\mathbf{f}\right)\right\}}{\sum_{\mathbf{e}'}\exp\left\{\boldsymbol{\lambda}\cdot\boldsymbol{h}\left(\mathbf{e}',\mathbf{f}\right)\right\}}, \tag{1}$$

where $\boldsymbol{h}$ represents features describing facets of $\mathbf{e}$ as a translation for $\mathbf{f}$ and $\boldsymbol{\lambda}$ represents the feature weights, constituting the model parameters.

To obtain estimates of the parameters, we use a tuning set, which consists of source sentences $\{\mathbf{f}\}_1^S$, their reference translations $\{\mathbf{r}\}_1^S$, and the candidate translations produced by a base translation system $\{\mathcal{E}\left(\mathbf{f}\right)\}_1^S$, each of which is in the form of an N-best list, i.e., $\mathcal{E}\left(\mathbf{f}_s\right)=\{\mathbf{e}_{s,1},...,\mathbf{e}_{s,N}\}$. To simplify our notation, we write $\mathcal{E}\left(\mathbf{f}_s\right)$ as $\mathcal{E}_s$ wherever no confusion will arise from doing so. The usual objective for optimization incorporates a task-specific evaluation metric:

$$\min_{\boldsymbol{\lambda}}\sum_{s=1}^{S}E\left(\mathbf{r}_s,\widehat{\mathbf{e}}\left(\mathbf{f}_s;\boldsymbol{\lambda}\right)\right), \tag{2}$$

where E is an error measure, which, in our case, is (one minus) BLEU [7], the evaluation metric to be used at test time.

Equation (2) is yet to be completed with a decision rule for selecting a proposed translation $\widehat{\mathbf{e}}\left(\mathbf{f}_s;\boldsymbol{\lambda}\right)$ from the pool of candidates $\mathcal{E}_s$. Traditionally, this decision rule is usually taken to be the so-called Maximum A Posteriori (MAP) decision rule:

$$\widehat{\mathbf{e}}\left(\mathbf{f}_s;\boldsymbol{\lambda}\right)=\arg\max_{\mathbf{e}\in\mathcal{E}_s}p\left(\mathbf{e}|\mathbf{f}_s;\boldsymbol{\lambda}\right)=\arg\max_{\mathbf{e}\in\mathcal{E}_s}\boldsymbol{\lambda}\cdot\boldsymbol{h}\left(\mathbf{e},\mathbf{f}_s\right). \tag{3}$$

Tuning is then achieved by optimizing (2) through minimum error rate training (MERT) [3], a fast, gradient-free optimization algorithm that has been widely used.

The same decision rule can be directly applied for decoding. Indeed, the MAP decision rule plays a role in both decoding and tuning in the traditional SMT landscape.

B. Minimum Bayes Risk Decoding

The MAP decision rule is arguably suboptimal. It is optimal with respect to the 0/1 loss,

$$L^{(0/1)}\left(\mathbf{r},\mathbf{e}\right)=\begin{cases}1 & \text{if } \mathbf{r}\neq\mathbf{e}\\ 0 & \text{otherwise,}\end{cases} \tag{4}$$

which is too harsh to be used in evaluation. For a general loss function, the optimal decision rule is given by the MBR decision rule [4]:

$$\widehat{\mathbf{e}}\left(\mathbf{f}_s;\boldsymbol{\lambda}\right)=\arg\min_{\mathbf{e}_i\in\mathcal{E}_s}\sum_{\mathbf{e}_j\in\mathcal{E}_s}L\left(\mathbf{e}_j,\mathbf{e}_i\right)p\left(\mathbf{e}_j|\mathbf{f}_s;\boldsymbol{\lambda}\right). \tag{5}$$

Clearly, if the loss function is taken to be the 0/1 loss, then the MBR decision rule reduces to the MAP rule.

The MBR decision rule is usually applied in decoding at test time to select a translation that supposedly minimizes the Bayes risk from among a pool of candidates. Kumar and Byrne [4] have found in their experiments that the MBR decision rule outperforms the MAP rule when the loss function is chosen to match the test-time evaluation metric, i.e., when $L=E$.

One may well wonder why this subject is worth further investigation, as the theory seems to have already identified

the *optimal* decision rule. However, the reality is not quite the same as the theory suggests. In practice, $\mathcal{E}_s$ in (5) never fully encompasses the entire candidate translation space. Furthermore, we can never be fully confident that we have provided an accurate model of $p(\mathbf{e}|\mathbf{f};\boldsymbol{\lambda})$. Indeed, these approximations prevent us from making any definitive statement regarding the optimality of the MBR rule. We therefore explore other possible decision rules, considering that an optimal decision rule is unlikely to exist in practical settings.

III. Listwise Ranking Functions

We begin by formulating log-linear models as ranking functions (Section III-A). This formulation is equivalent to the usual probabilistic view, but it can be naturally extended to ranking functions that consider the entire N-best list, known as listwise ranking functions (Section III-B). From this basis, we discover that our formulation can be directly embedded back into the log-linear model by treating the listwise information as features (Section III-C) and is therefore compatible with the MERT algorithm [3].

A. Log-linear Models as Ranking Functions

For the settings considered in Section II-A, particularly for the objective expressed in (2) and the MAP decision rule (3), the log-linear model can be equivalently regarded as a linear ranking function:

$$f(\mathbf{e}|\mathbf{f};\boldsymbol{\lambda}) = \boldsymbol{\lambda} \cdot \boldsymbol{h}(\mathbf{e},\mathbf{f}). \tag{6}$$

This ranking function is used to derive an ordering over the N-best list $\mathcal{E}$. With the MAP decision rule, only the top-ranked candidate that is assigned the highest value in the list by the ranking function is selected. Because only the ordering matters, the magnitude of $\boldsymbol{\lambda}$ is irrelevant. Moreover, any monotonic transformation, such as $\exp$, would not affect the ordering.

This observation has been noted in previous work [5], where the function used to derive the ordering was called a scorer. We instead term it a ranking function, to highlight its rank-determining nature and to use it as a stepping stone toward our listwise extension.

B. From Pointwise Ranking Functions to Listwise

The above ranking function (6) is called *pointwise* because it scores each item $\mathbf{e}$ without considering the other items in the N-best list $\mathcal{E}$. However, such a ranking function is restrictive for the task at hand. Minimizing objective (2) calls for a listwise focus because it is equivalent to maximizing NDCG@1 [8] with the gain function $1-E$, whereas Ravikumar *et al.* [9] have shown that the Bayes-consistent ranking function of the NDCG metric is inherently *listwise*.

A listwise ranking function scores the N-best list $\mathcal{E}$ as a whole, returning a vector of scores to be sorted to yield the ordering. For SMT, the desired listwise ranking function takes the form

$$\boldsymbol{F}(\mathcal{E};\boldsymbol{\theta}) = [F_1(\mathcal{E};\boldsymbol{\theta}), \ldots, F_i(\mathcal{E};\boldsymbol{\theta}), \ldots, F_N(\mathcal{E};\boldsymbol{\theta})],$$

where the subscript of F, $i \in \{1,...,N\}$, indexes the score to be assigned to $\mathbf{e}_i$, the i-th candidate in $\mathcal{E}$.

It is evident that pointwise ranking functions can be regarded as a special case of functions of this form. For such a ranking function, we can write our decision rule as

$$\widehat{i}(\mathbf{f};\boldsymbol{\theta}) = \arg\max_{i\in\{1,\ldots,N\}} F_i(\mathcal{E};\boldsymbol{\theta}),$$

which will reduce to the MAP decision rule (3) if we use a pointwise ranking function (6). Recall that $\mathcal{E}$ implicitly depends on the source sentence $\mathbf{f}$.

Although desirable, the listwise formulation of a ranking function is cumbersome to work with because of the difficulty of encoding the interactions among a list of N items. Fortunately, Pareek and Ravikumar [6] recently addressed this problem by assuming a natural exchangeability condition to obtain a compact representation in the following form:

$$F_i(\mathcal{E};\boldsymbol{\theta}) = \sum_t \prod_{j\neq i}^{N} g_t(\mathbf{e}_i, \mathbf{e}_j;\boldsymbol{\theta}). \tag{7}$$

This theory suggests that a listwise ranking function can be constructed from appropriate pairwise functions g.

C. Listwise Ranking Functions for Statistical Machine Translation

Before proceeding, let us first examine the validity of the exchangeability assumption for SMT. Intuitively, exchangeability formalizes the notion that ranking functions should depend only on the features of items, with their positions in the N-best list being irrelevant. This notion can be more formally defined as follows [6].

Definition 1. (Exchangeability) A listwise ranking function $\boldsymbol{F}$ is said to be exchangeable if $\boldsymbol{F}(\pi(\mathcal{E})) = \pi(\boldsymbol{F}(\mathcal{E}))$ for every permutation π that operates on N elements.

This is indeed a natural condition, because we wish to use the function values to induce an ordering over the candidate space and the induced ordering should be unaffected by the original arbitrary ordering presented to the function. For SMT in particular, we would like the ranking function to assign the highest score to the best translation, regardless of the position in which it may originally reside in the N-best list. Therefore, the form of (7) yields the family of listwise ranking functions that we desire.

Given the listwise representation in (7), we must decide on the parametrization of the pairwise functions. We follow Pareek and Ravikumar [6] in using

$$F_i(\mathcal{E};\boldsymbol{\theta}) = b(\mathbf{e}_i,\mathbf{f};\boldsymbol{\lambda}) \prod_{j\neq i}^{N} \exp\{\boldsymbol{\mu} \cdot \boldsymbol{S}(\mathbf{e}_i,\mathbf{e}_j,\mathbf{f})\}, \tag{8}$$

with model parameters $\boldsymbol{\theta} = \{\boldsymbol{\lambda},\boldsymbol{\mu}\}$. This expression is obtained from (7) by taking only a single term from the series and taking the pairwise function g to be an exponential in the pairwise function $\boldsymbol{S}$ weighted by $\boldsymbol{\mu}$, with a base function b.

In SMT, we are inclined to choose the base function to be the traditional (exponentiated) pointwise ranking function (6):

$$b(\mathbf{e},\mathbf{f};\boldsymbol{\lambda}) = \exp\{\boldsymbol{\lambda} \cdot \boldsymbol{h}(\mathbf{e},\mathbf{f})\}.$$

Therefore, (8) becomes

$$F_i\left(\mathcal{E};\boldsymbol{\theta}\right)=\exp\left\{\boldsymbol{\lambda}\cdot\boldsymbol{h}\left(\mathbf{e}_i,\mathbf{f}\right)+\boldsymbol{\mu}\cdot\sum_{j\neq i}^{N}\boldsymbol{S}\left(\mathbf{e}_i,\mathbf{e}_j,\mathbf{f}\right)\right\},$$

which, as a ranking function, is equivalent to

$$F_i\left(\mathcal{E};\boldsymbol{\theta}\right)=\boldsymbol{\lambda}\cdot\boldsymbol{h}\left(\mathbf{e}_i,\mathbf{f}\right)+\boldsymbol{\mu}\cdot\sum_{j\neq i}^{N}\boldsymbol{S}\left(\mathbf{e}_i,\mathbf{e}_j,\mathbf{f}\right). \quad (9)$$

Thus, the task of developing listwise ranking functions is reduced to that of designing the pairwise function $\boldsymbol{S}$, which can take any form with output in any number of dimensions, as long as we expect that form to be useful.

The form of (9) is reminiscent of the pointwise ranking function (6), except that it is augmented with new *listwise features* that gather information from the entire N-best list through the pairwise function $\boldsymbol{S}$, with corresponding weights $\boldsymbol{\mu}$. Therefore, the popular MERT algorithm is directly applicable to our formulation, allowing the joint tuning of $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$. This is a distinctive advantage offered by our formulation, whereas in the original form (8) as used in [6], the parameters $\boldsymbol{\lambda}$ of the base function must be tuned separately beforehand and then held fixed to proceed with the tuning of $\boldsymbol{\mu}$. We will show that the joint tuning of the parameters is essential to the performance improvement achieved using our approach (Section IV-D).

D. Minimum Bayes Risk Decoding as a Special Case

Within our formulation, the MBR decision rule can be recovered as a special case, which we demonstrate as follows.

We begin with the MBR decision rule (5), where we take the candidate translation space $\mathcal{E}$ to be an N-best list and use $\widehat{i}$ to index $\widehat{\mathbf{e}}\left(\mathbf{f};\boldsymbol{\lambda}\right)$:

$$\begin{aligned}&\widehat{i}\left(\mathbf{f};\widetilde{\boldsymbol{\lambda}}\right)\\&=\arg\min_i\sum_{j=1}^{N}L\left(\mathbf{e}_j,\mathbf{e}_i\right)p\left(\mathbf{e}_j|\mathbf{f};\widetilde{\boldsymbol{\lambda}}\right)\\&=\arg\max_i\sum_{j\neq i}^{N}G\left(\mathbf{e}_j,\mathbf{e}_i\right)\exp\left\{\widetilde{\boldsymbol{\lambda}}\cdot\boldsymbol{h}\left(\mathbf{e}_j,\mathbf{f}\right)\right\}.\end{aligned}$$

In this derivation, we have used $L\left(\mathbf{e},\mathbf{e}\right)=0$ and defined a gain function $G\left(\mathbf{e}_j,\mathbf{e}_i\right)=1-L\left(\mathbf{e}_j,\mathbf{e}_i\right)$. Note that in this case, the magnitudes of the parameters matter, which we indicate with a tilde. This decision rule can also be treated as a ranking function:

$$F_i^{(\mathrm{MBR})}\left(\mathcal{E}|\mathbf{f};\widetilde{\boldsymbol{\lambda}}\right)=\sum_{j\neq i}^{N}G\left(\mathbf{e}_j,\mathbf{e}_i\right)\exp\left\{\widetilde{\boldsymbol{\lambda}}\cdot\boldsymbol{h}\left(\mathbf{e}_j,\mathbf{f}\right)\right\}. \quad (10)$$

Comparing this with our ranking function (9), we recognize that

$$\begin{aligned}\boldsymbol{\lambda}&=\mathbf{0}\\\boldsymbol{S}\left(\mathbf{e}_i,\mathbf{e}_j,\mathbf{f}\right)&=G\left(\mathbf{e}_j,\mathbf{e}_i\right)\exp\left\{\widetilde{\boldsymbol{\lambda}}\cdot\boldsymbol{h}\left(\mathbf{e}_j,\mathbf{f}\right)\right\}\end{aligned}$$

would reduce (9) to (10). Note that $\boldsymbol{S}$ is taken to be a scalar. This reflects the flexibility of our formulation in that it allows more than one gain function to be included.

E. Features Derived from the Pairwise Function

The augmented listwise features to be added to the log-linear model are fully specified by the pairwise function used. They could, in principle, depend on the source sentence $\mathbf{f}$, but such features would be difficult to design; therefore, we neglect this dependence and hope that the baseline features $\boldsymbol{h}$ can sufficiently capture this information. The new features that we investigate in this paper are primarily motivated by various evaluation metrics, as listed below.

- Sentence-level BLEU: This is intended as an approximation to the test-time evaluation metric, the corpus-level BLEU. We implement it following [10], except that we do not scale the reference length.
- Word error rate (WER): The string edit distance divided by the number of words in the reference sentence [3].
- Translation edit rate (TER): An extension to the WER that further allows a phrasal shift as a legitimate edit operation [11].
- Meteor: A sentence-level metric that considers stemming and word order [12].
- Cosine similarity between $\boldsymbol{h}\left(\mathbf{e}_i,\mathbf{f}\right)$ and $\boldsymbol{h}\left(\mathbf{e}_j,\mathbf{f}\right)$.

Because the number of candidates may differ among different source sentences, we also include an averaged variant of each feature above.

These choices for the pairwise function involve complex dependence on the translations, which prohibits their integration into the decoder. We therefore execute our approach within the discriminative reranking framework [13]. A visualization of our system architecture is presented in Fig. 2.

IV. Experiments

A. Experimental Settings

We conducted our experiments on Chinese-English translation. Our training set consisted of 1.23M parallel sentences containing 32.1M Chinese words and 35.4M English words. We used a 4-gram language model trained on the Xinhua portion of the English GIGAWORD corpus (398.6M words). We used the NIST 2006 MT Chinese-English data set as the development set and the NIST 2002-2005 and 2008 MT Chinese-English data sets as the test sets. All corpora were lowercased and tokenized. We measured the system performance using the case-insensitive BLEU-4 score. Statistical significance testing was performed with paired bootstrap resampling [14]. One or two symbols following the reported BLEU score indicate a significance level of $p<0.05$ or $p<0.01$, respectively.

We implemented our baseline using Moses [15], a phrase-based machine translation toolkit. However, as an extension to log-linear models, our approach can be applied to any system that produces N-best lists. Our baseline features included 4 translation models (phrase translation probability and lexical weighting, for both translation directions), a language model, a word penalty, a phrase penalty, and 7 reordering models (bidirectional `msd` models and a distance model), for a total of 14 features.

After obtaining the N-best lists, we augmented them with our new features. Their weights were tuned on the development set using Z-MERT [16] and then used to rerank the N-

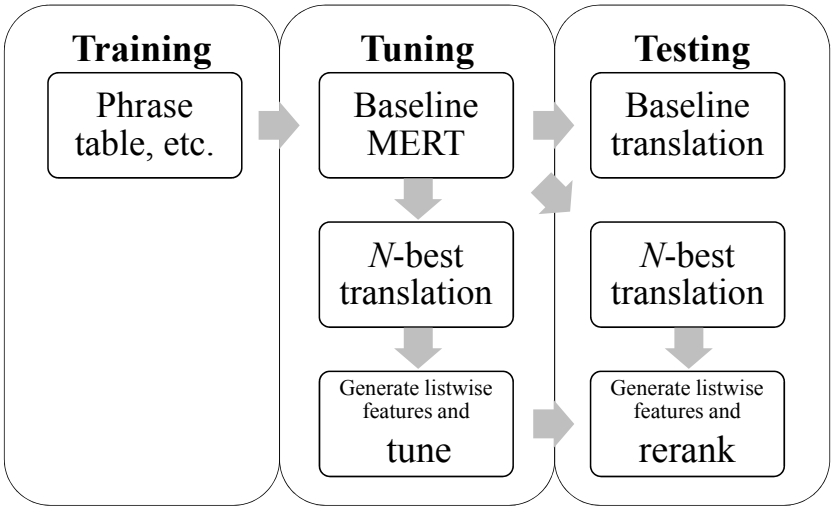


Figure 2. System architecture. The core steps of our approach involve generating listwise features for the joint tuning of the model parameters and then again at test time for reranking.

Table I
EFFECTS OF LISTWISE FEATURES

Listwise feature	Dev	MT 02	MT 03	MT 04	MT 05	MT 08
sBLEU	30.70	32.59	32.61	32.80	31.15	24.66
WER_avg	30.58	32.58	32.23	32.66	30.91	24.90
TER_avg	30.64	32.75	32.25	32.86	31.09	24.93
Meteor_avg	30.66	32.44	32.10	32.66	31.09	24.78
sBLEU + Meteor_avg (default)	**30.76**	**32.87**	32.44	33.00	31.38	**25.01**
sBLEU + WER_avg + TER_avg + Meteor_avg	30.70	32.82	**32.63**	**33.02**	**31.44**	24.69

Table II
COMPARISON WITH BASELINE AND MBR APPROACHES

System	Dev	MT 02	MT 03	MT 04	MT 05	MT 08
Baseline	30.34	32.35	31.92	32.58	30.85	24.61
MBR	30.50	32.71	32.13	32.72	30.92	25.10
Our system	30.76^{**++}	32.87^{**+}	32.44^{**++}	33.00^{**++}	31.38^{**++}	25.01**

best lists for the test sets. These steps are illustrated in Fig. 2.

We used $N = 1000$ throughout our experiments unless otherwise noted (Sections IV-F and IV-G). Following [5], we removed duplicate translations from the N-best lists before computing the listwise features for our system; this procedure has the additional benefit of reducing computation time.

B. Feature Choice

In this section, we elaborate on our feature choice and examine the relative effectiveness of the various listwise features. Choosing among a pool of features is a non-trivial issue because of the instability of the MERT algorithm, as introducing irrelevant features into the model is likely to have a detrimental effect [17]. Therefore, we followed the approach used in [13]. We first added each of these features individually to the baseline approach to observe the resulting improvements over the baseline. Based on the results, we were able to determine cosine similarity to be a useless pairwise function and excluded it from subsequent experiments. All other features were found to be beneficial to a certain degree. Then, we combined pairs of potential features, during which process we found there to be no benefit in combining a pairwise function with its averaged variant, likely because of the strong correlation between them. This step revealed the most powerful feature combination in our experiments, the sentence-level BLEU plus the averaged Meteor. The results obtained using the various combinations highlight the benefit of incorporating more than one pairwise function, which is impossible in the MBR approach. Further combining of functions yielded little gain, likely again because of the increasing overlap between the features and the fact that increasing the number of features may diminish the benefit that those additional features could provide [17]. These findings are summarized in Table I, where a suffix of avg in a feature name indicates an averaged variant and bold typeface indicates the best figure achieved for a given data set.

We also found that when a single listwise feature is used,

Position	Original N-best list	System
1	Australia to **open** embassy in Manila	Baseline
2	Australia to **reopen** embassy in Manila	
3	The Australian embassy in Manila **reopens**	
4	Australian embassy in Manila **reopens**	
5	The Australian embassy in Manila **reopened**	Our system
...	...	
15	The Australian embassy in Manila to **open**	MBR
	Australia **reopened** Manila embassy	Reference

Figure 3. A real example from the test set that demonstrates the benefit of reranking with the listwise features produced by our system. The sentences have been truecased for presentation. The last row shows one of the four reference translations. Bold typeface signifies the words that make a semantic difference in the translations.

Table III
EFFECT OF JOINT TUNING

System	Dev	MT 02	MT 03	MT 04	MT 05	MT 08
Baseline	30.34	32.35	31.92	32.58	30.85	24.61
No joint tuning	30.35	32.34	31.95	32.58	30.85	24.65
Joint tuning	30.76**	32.87**	32.44**	33.00**	31.38**	25.01**

Table IV
COMPARISON WITH POINTWISE RERANKING

System	Dev	MT 02	MT 03	MT 04	MT 05	MT 08
Pointwise reranker	30.52	32.79	32.10	32.50	31.42	24.53
Pointwise reranker + listwise features	33.12**	33.00	32.98**	33.00**	31.83**	25.11**

similarity functions (sentence-level BLEU and Meteor) are assigned positive weights, whereas distance functions (WER and TER) are assigned negative ones. This phenomenon may reflect a trend in N-best lists that good translations are often similar to their alternatives. However, this trend sometimes becomes blurred when more listwise features are added, possibly because complex interactions between these features obscure the signs of the weights.

In subsequent tests, we used the sentence-level BLEU plus the averaged Meteor as our default set of listwise features.

C. Comparison with Baseline and Minimum Bayes Risk Decoding

In this section, we report results obtained using our system and the baseline. We additionally compare the results with those obtained via MBR decoding using the implementation in Moses. To determine the scaling factor that controls the magnitude of $\widetilde{\boldsymbol{\lambda}}$ in the MBR method (Equation (10)), we performed a grid search on the development set [18].

Table II summarizes the results. Our system significantly improves on the baseline for all data sets (marked with asterisks), with a consistent improvement ranging from 0.40 to 0.53. It also significantly improves on the MBR approach for most data sets (marked with plus signs), with the exception that it exhibits comparable performance on MT 08, which behaves like an outlier, with a considerably lower BLEU compared with the others. The MT 08 anomaly can likely be attributed to a domain difference, because unlike the MT 02 - 05 sets, which comprise news data, MT 08 contains a considerable amount of informal language found on web forums.

As a concrete counterpart to the illustrative example presented in Fig. 1, Fig. 3 shows a case in which our system successfully uses listwise information to identify a better translation candidate. The output of our system is originally ranked fifth in the list. In fact, all candidates in the top-five list, except the baseline output that is originally ranked at the top, use a form of "reopen" in their translations. This similarity, in turn, is reflected in higher listwise scores in the output of our system. The MBR method fails to discover this trend in the list, possibly because of its inability to cope with the various word forms in the absence of the Meteor feature.

D. Effect of Joint Tuning

Intuitively, joint tuning should exert a beneficial effect on the system performance. We validated this intuition by testing a system with joint tuning disabled. This was achieved by fixing the 14 baseline feature weights and allowing the MERT algorithm to adjust only the weights of the listwise features. This mirrors the experimental setting considered in [6], where joint tuning was unavailable.

The BLEU scores obtained using this system are reported in Table III, alongside the corresponding results for the baseline

Table V
EFFECT OF THE SIZE OF THE N-BEST LIST

N	System	Dev	MT 02	MT 03	MT 04	MT 05	MT 08
100	Baseline	30.27	32.44	32.16	32.45	31.04	24.41
	Reranker	30.57**	32.32	32.39**	32.45	31.02	24.78**
	Improvement	+0.30	-0.12	+0.23	+0.00	-0.02	+0.37
500	Baseline	30.49	32.28	31.92	32.39	30.79	24.41
	Reranker	30.67*	32.28	32.23**	32.57**	30.90	24.66**
	Improvement	+0.18	+0.00	+0.31	+0.18	+0.11	+0.25
1000	Baseline	30.34	32.35	31.92	32.58	30.85	24.61
	Reranker	30.76**	32.87**	32.44**	33.00**	31.38**	25.01**
	Improvement	+0.42	+0.52	+0.53	+0.42	+0.53	+0.40

system and the version of our system with joint tuning. We observe that the system without joint tuning performs only comparably to the baseline, whereas the version with joint tuning significantly surpasses it (marked with asterisks). We conjecture that joint tuning opens up more parameter regions for the MERT algorithm to explore, thereby providing the opportunity to find better local optima. This result matches our intuition and confirms that the joint tuning capability, which could not be provided by previously developed systems, is critical to our task.

E. Comparison with Traditional Discriminative Reranking

Because our model falls into the framework of discriminative reranking, we also compared its results with those of a model that performs reranking based on pointwise features. More specifically, we investigated the improvement achieved by using our listwise features on top of an existing reranker. The pointwise features that we used were 4- and 5-gram language models trained on a superset of the baseline language model corpus, with the addition of the AFP portion of the English GIGAWORD corpus, amounting to 1254.8M words.

Table IV presents the additional gain achieved by a traditional reranker using our listwise features. The improvements are significant for most data sets (marked with asterisks), and the largest absolute increase is near 0.9. This implies that our listwise features are complementary to these pointwise features and provide additional information that is valuable to the reranker.

F. Effect of the Size of the N-best List

Clearly, the choice of N has an impact on our model. A larger N results in the availability of more candidates for reranking and affects the computation of the features.

It is worth noting that the numbers of candidates used for tuning and testing could be different. However, we found in our preliminary experiments that a matched size yields the best performance, which is intuitively reasonable. Therefore, we adhered to $N_{\text{dev}} = N_{\text{test}} = N$ in our experiments. Notably, the N value that we report is an upper bound specified to the decoder to truncate its output. The post-processing applied to remove duplicates approximately halved that number.

We varied the candidate size N on the set $\{100, 500, 1000\}$. Table V summarizes the results of the comparison. We observe that going from 100 to 500 results in more stable improvement over the baseline and that the improvement gap widens from 500 to 1000. This indicates that exploring larger N values will typically be beneficial when seeking better performance.

G. Running Time

The most time-demanding step of our approach is the computation of the listwise features. We timed the computation of the features derived from the sentence-level BLEU and the string edit distance, whose own complexities depend on the length of the candidates and contribute a multiplicative factor to the overall complexity. We performed the experiments on a 2.6 GHz Linux machine.

As seen from (9), the time complexity of computation scales quadratically with the size of the candidate space $\mathcal{E}$, or as $O\left(N^2\right)$ if we assume an upper bound N on the size to simplify the discussion. This is verified by Fig. 4. Note that when producing this figure, we have not removed duplicates to avoid distorting the observed trend. As a more practical time value, for $N = 1000$ with deduplication enabled, our vanilla implementation required approximately one minute to compute the features for all candidates for each source sentence. This number is an average over all data sets.

The same complexity issue is encountered in the MBR approach, which also scales quadratically. The problem can be alleviated through parallelization because the independence between source sentences makes this level of parallelization trivial.

V. RELATED WORK

Our work is related to three lines of research, as discussed below.

A. Minimum Bayes Risk and Consensus Approach

As noted before, other decision rules exist that also consider the entire N-best list [4], [5], but they lack a mechanism for combining multiple evaluation metrics that are potentially useful. The ideas can be ported to a tuning objective [3], [19] or used to invent an ad hoc objective [20]. By contrast, we treat listwise information as features with weights to be tuned and adhere to the objective (2) that has proven effective.

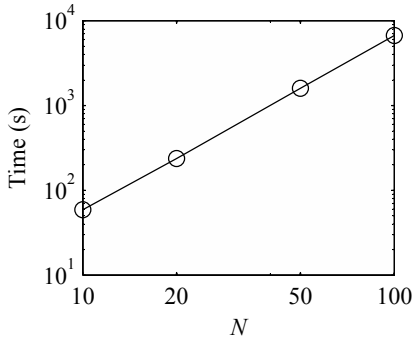


Figure 4. Running time in seconds versus N, in a log-log plot, for the generation of the sentence-level BLEU and string edit distance features on MT 08 without the removal of duplicate candidates in the N-best lists. These findings reveal the quadratic time complexity of the feature computation with respect to N.

Tromble *et al.* [21] extended MBR decoding from N-best lists to lattices, a topic that is out of the scope of this paper. However, to permit an efficient implementation, they approximated their gain function log(BLEU) using a linear decomposition and set the linear weights heuristically. Kumar *et al.* [18] further estimated these linear weights using MERT. These techniques are similar in form to our approach, although they also differ in a few ways. First, they are conceived as an approximation to the MBR method with the goal of efficiency, whereas our approach is motivated from the ranking perspective as a different decision rule. Second, the pairwise functions used in the linear approximation are no longer any particular evaluation metric, whereas we explore combinations of commonly used evaluation metrics. Finally, we can view these combinations of metrics as particular instances of our pairwise function $\boldsymbol{S}$ (cf. Equations (9) and (10)).

B. Learning to Rank

Learning to rank is a topic in machine learning with a vast body of literature; see, e.g., [22] for a survey.

It is worthwhile to distinguish between a listwise *ranking function* and a listwise *tuning objective* at this point. Traditionally, learning-to-rank methods have been divided into pointwise, pairwise and listwise methods [22], but this distinction is made with respect to the adopted tuning objective, and listwise objectives that optimize toward task-specific metrics such as the NDCG have generally been deemed more effective. By contrast, whether the ranking function should be pointwise or listwise (local or global, in the terminology of [22]) has been less extensively discussed. Indeed, prior to the work of [6], most ranking functions were chosen to be pointwise, with only a few exceptions [23]–[27].

For SMT, the traditional log-linear model optimized toward the target metric (Section II-A) can be regarded as a pointwise ranking function with a listwise tuning objective.

Our approach builds upon the work of [6], which advocates listwise ranking functions. We seamlessly join the listwise representation for ranking functions with the existing log-linear framework, and we have shown that this tight integration opens up the possibility to jointly tune the listwise feature weights along with the traditional pointwise ones, which is a property that is important for our task. Previous works in which listwise ranking functions have been utilized either lack this joint tuning property [6] or do not directly optimize a listwise tuning objective [24]–[26], or both [23], [27].

Learning to rank has found successful application in information retrieval [28, *inter alia*]. However, efforts to translate its effectiveness to machine translation have borne much less fruit. For example, the published results in [29] and [30] are mixed compared with those of MERT. Their ranking functions are pointwise, as usual. This may suggest that MERT is already quite effective in optimizing our objective (2), and thus, it wins over many techniques developed to optimize surrogate objectives in learning to rank.

We note in passing that pairwise ranking optimization (PRO) [31] is a technique developed to optimize a tuning objective in the context of machine translation with a learning-to-rank focus. As a tuning algorithm, it is orthogonal to our approach, which focuses on ranking functions.

C. Discriminative Reranking

A vast body of literature on discriminative reranking also exists in the field of machine translation [13, *inter alia*] and in the broader field of natural language processing [32, *inter alia*]. Such work is generally aimed at finding features that better describe the relationship between a target translation and a source sentence (in a pointwise fashion). These features often have linguistic motivations. By contrast, our work does not attempt to develop such features in the first place. Indeed, the pairwise functions explored in this paper only loosely relate target translations to their source sentences. Rather, they relate candidate translations to each other. We have confirmed in our experiment that our listwise features provide a benefit that is complementary to that of traditional pointwise features (Section IV-E). Nevertheless, such work and ours both fit within the log-linear framework and take advantage of the effectiveness of MERT.

One notable exception is the work of Ueffing and Ney [33]. Their work focuses on word-level confidence estimation, but sentence-level confidence can be obtained by multiplying the word-level confidence estimates and can then be used as an additional model for reranking. Of particular relevance are their system-based approaches to word-level confidence estimation, in which statistics are computed based on translation system output, e.g., an N-best list. However, these authors do not report the reranking performance of their system-based approaches. In essence, the system-based sentence-level confidence represents a heuristic means of harvesting listwise information outside the theoretical framework of listwise ranking functions described in Section III-B. Admittedly, there are numerous possible methods of devising heuristics, and we attempted several in our preliminary experiments, but they did not prove useful; this experience illustrates the value of theoretical guidance.

VI. Conclusion

In this paper, we view log-linear models in statistical machine translation from a ranking perspective. This viewpoint

admits a natural extension to the case in which the entire N-best list is considered, through a shift from the usual pointwise ranking functions to listwise ranking functions. Such a formulation proves to be easily incorporated into the existing log-linear framework, with an extension that allows any number of evaluation metrics or other pairwise functions to be included in the model, and hence encompasses the minimum Bayes risk approach as a special case. We have identified several useful pairwise functions in our experiments that demonstrate significant improvement on the usual log-linear model.

Another technical contribution of this paper lies in the presentation of a model that allows the joint tuning of pointwise and listwise feature weights, which is a desirable property but was not previously possible under a listwise tuning objective.

In future work, it may be worthwhile to explore other types of pairwise functions, such as those that arise in the case of various sparse features. We are also interested in the interactions between our model and other tuning algorithms, such as PRO and MIRA [34].

REFERENCES

[1] P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer, "The Mathematics for Statistical Machine Translation: Parameter Estimation," *Computational Linguistics*, 1993.

[2] F. J. Och and H. Ney, "Discriminative Training and Maximum Entropy Models for Statistical Machine Translation," in *Proceedings of the 40th Annual Meeting of the ACL*, 2002. [Online]. Available: http://dx.doi.org/10.3115/1073083.1073133

[3] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in *Proceedings of the 41st Annual Meeting of the ACL*, 2003. [Online]. Available: http://dx.doi.org/10.3115/1075096.1075117

[4] S. Kumar and W. Byrne, "Minimum Bayes-Risk Decoding for Statistical Machine Translation," in *Proceedings of HLT-NAACL*, 2004.

[5] J. DeNero, D. Chiang, and K. Knight, "Fast Consensus Decoding over Translation Forests," in *Proceedings of the Joint Conference of ACL/AFNLP*, 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1690219.1690226

[6] H. H. Pareek and P. K. Ravikumar, "A Representation Theory for Ranking Functions," in *Advances in Neural Information Processing Systems 27*, 2014. [Online]. Available: http://papers.nips.cc/paper/5250-a-representation-theory-for-ranking-functions.pdf

[7] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 40th Annual Meeting of the ACL*, 2002. [Online]. Available: http://dx.doi.org/10.3115/1073083.1073135

[8] K. Järvelin and J. Kekäläinen, "Cumulated Gain-based Evaluation of IR Techniques," *ACM Transactions on Information Systems*, 2002. [Online]. Available: http://doi.acm.org/10.1145/582415.582418

[9] P. D. Ravikumar, A. Tewari, and E. Yang, "On NDCG consistency of listwise ranking methods," in *International Conference on Artificial Intelligence and Statistics*, 2011.

[10] X. He and L. Deng, "Maximum Expected BLEU Training of Phrase and Lexicon Translation Models," in *Proceedings of the 50th Annual Meeting of the ACL*, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2390524.2390566

[11] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and R. Weischedel, "A Study of Translation Error Rate with Targeted Human Annotation," in *Proceedings of AMTA*, 2006.

[12] M. Denkowski and A. Lavie, "Meteor Universal: Language Specific Translation Evaluation for Any Target Language," in *Proceedings of the Ninth Workshop on Statistical Machine Translation*, 2014. [Online]. Available: http://aclanthology.info/papers/meteor-universal-language-specific-translation-evaluation-for-any-target-language

[13] F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev, "A Smorgasbord of Features for Statistical Machine Translation," in *Proceedings of HLT-NAACL*, 2004.

[14] P. Koehn, "Statistical Significance Tests for Machine Translation Evaluation," in *Proceedings of EMNLP*, 2004. [Online]. Available: http://aclanthology.info/papers/statistical-significance-tests-for-machine-translation-evaluation

[15] 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 ACL on Interactive Poster and Demonstration Sessions*, 2007. [Online]. Available: http://dl.acm.org/citation.cfm?id=1557769.1557821

[16] O. Zaidan, "Z-MERT: A Fully Configurable Open Source Tool for Minimum Error Rate Training of Machine Translation Systems," *The Prague Bulletin of Mathematical Linguistics*, 2009. [Online]. Available: http://www.degruyter.com/view/j/pralin.2009.91.issue–1/v10108-009-0018-2/v10108-009-0018-2.xml

[17] D. Chiang, Y. Marton, and P. Resnik, "Online Large-margin Training of Syntactic and Structural Translation Features," in *Proceedings of EMNLP*, 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1613715.1613747

[18] S. Kumar, W. Macherey, C. Dyer, and F. Och, "Efficient Minimum Error Rate Training and Minimum Bayes-Risk Decoding for Translation Hypergraphs and Lattices," in *Proceedings of the Joint Conference of ACL/AFNLP*, 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1687878.1687903

[19] D. A. Smith and J. Eisner, "Minimum Risk Annealing for Training Log-linear Models," in *Proceedings of COLING/ACL*, 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1273073.1273174

[20] A. Pauls, J. DeNero, and D. Klein, "Consensus Training for Consensus Decoding in Machine Translation," in *Proceedings of EMNLP*, 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1699648.1699688

[21] R. W. Tromble, S. Kumar, F. Och, and W. Macherey, "Lattice Minimum Bayes-Risk Decoding for Statistical Machine Translation," in *Proceedings of EMNLP*, 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1613715.1613792

[22] H. Li, "Learning to Rank for Information Retrieval and Natural Language Processing," *Synthesis Lectures on Human Language Technologies*, 2011.

[23] T. Qin, T.-Y. Liu, X.-D. Zhang, D.-S. Wang, W.-Y. Xiong, and H. Li, "Learning to Rank Relational Objects and Its Application to Web Search," in *Proceedings of the 17th International Conference on World Wide Web*, ser. WWW '08. New York, NY, USA: ACM, 2008, pp. 407–416. [Online]. Available: http://doi.acm.org/10.1145/1367497.1367553

[24] T. Qin, T.-y. Liu, X.-d. Zhang, D.-s. Wang, and H. Li, "Global Ranking Using Continuous Conditional Random Fields," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Curran Associates, Inc., 2008, pp. 1281–1288. [Online]. Available: http://papers.nips.cc/paper/3402-global-ranking-using-continuous-conditional-random-fields.pdf

[25] M. N. Volkovs and R. S. Zemel, "BoltzRank: Learning to Maximize Expected Ranking Gain," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: ACM, 2009, pp. 1089–1096. [Online]. Available: http://doi.acm.org/10.1145/1553374.1553513

[26] S. Ji, K. Zhou, C. Liao, Z. Zheng, G.-R. Xue, O. Chapelle, G. Sun, and H. Zha, "Global Ranking by Exploiting User Clicks," in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '09. New York, NY, USA: ACM, 2009, pp. 35–42. [Online]. Available: http://doi.acm.org/10.1145/1571941.1571950

[27] C. Kang, X. Wang, J. Chen, C. Liao, Y. Chang, B. Tseng, and Z. Zheng, "Learning to Re-rank Web Search Results with Multiple Pairwise Features," in *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, ser. WSDM '11. New York, NY, USA: ACM, 2011, pp. 735–744. [Online]. Available: http://doi.acm.org/10.1145/1935826.1935924

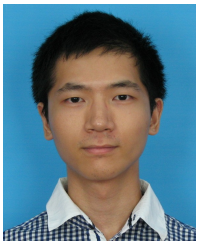
[28] T. Joachims, "Optimizing Search Engines Using Clickthrough Data," in *Proceedings of the Eighth ACM SIGKDD*, 2002. [Online]. Available: http://doi.acm.org/10.1145/775047.775067

[29] L. Shen, A. Sarkar, and F. J. Och, "Discriminative Reranking for Machine Translation," in *Proceedings of HLT-NAACL*, 2004.

[30] K. K. Duh, "Learning to rank with partially-labeled data," Ph.D. dissertation, University of Washington, 2009.

[31] M. Hopkins and J. May, "Tuning As Ranking," in *Proceedings of EMNLP*, 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=2145432.2145575

[32] M. Collins and T. Koo, "Discriminative Reranking for Natural Language Parsing," *Computational Linguistics*, 2005. [Online]. Available: http://dx.doi.org/10.1162/0891201053630273
[33] N. Ueffing and H. Ney, "Word-Level Confidence Estimation for Machine Translation," *Computational Linguistics*, vol. 33, no. 1, pp. 9–40, 2007. [Online]. Available: http://dx.doi.org/10.1162/coli.2007.33.1.9
[34] K. Crammer and Y. Singer, "Ultraconservative Online Algorithms for Multiclass Problems," *The Journal of Machine Learning Research*, 2003. [Online]. Available: http://dx.doi.org/10.1162/jmlr.2003.3.4-5.951



Meng Zhang is a Ph.D. student in the Department of Computer Science and Technology at Tsinghua University. His current research interests include natural language processing and machine translation.

Yang Liu is an Associate Professor in the Department of Computer Science and Technology at Tsinghua University. His current research interests include natural language processing and machine translation.

Huanbo Luan is the deputy executive director of NExT Search Center at both Tsinghua University and National University of Singapore. His research interests include multimedia information retrieval, social media, and big data analysis.

Maosong Sun is a Full Professor in the Department of Computer Science and Technology at Tsinghua University. His current research interests include natural language processing, Chinese information processing, and computational social science.