Minimum Bayes Risk based Answer Re-ranking for Question Answering

Nan Duan
Natural Language Computing
Microsoft Research Asia
nanduan@microsoft.com

Abstract

This paper presents two minimum Bayes risk (MBR) based Answer Re-ranking (**MBRAR**) approaches for the question answering (QA) task. The first approach re-ranks single QA system's outputs by using a traditional MBR model, by measuring correlations between answer candidates; while the second approach re-ranks the combined outputs of multiple QA systems with heterogenous answer extraction components by using a mixture model-based MBR model. Evaluations are performed on factoid questions selected from two different domains: Jeopardy! and Web, and significant improvements are achieved on all data sets.

1 Introduction

Minimum Bayes Risk (MBR) techniques have been successfully applied to a wide range of natural language processing tasks, such as statistical machine translation (Kumar and Byrne, 2004), automatic speech recognition (Goel and Byrne, 2000), parsing (Titov and Henderson, 2006), etc. This work makes further exploration along this line of research, by applying MBR technique to question answering (QA).

The function of a typical factoid question answering system is to automatically give answers to questions in most case asking about entities, which usually consists of three key components: question understanding, passage retrieval, and answer extraction. In this paper, we propose two *MBR-based Answer Re-ranking (MBRAR)* approaches, aiming to re-rank answer candidates from either single and multiple QA systems. The first one re-ranks answer outputs from single QA system based on a traditional MBR model by measuring the correlations between each answer candidates and all the other candidates; while the second one re-ranks the combined answer outputs from multiple QA systems based on a mixture model-based MBR model. The key contribution of this work is that, our MBRAR approaches assume little about QA systems and can be easily applied to QA systems with arbitrary sub-components.

The remainder of this paper is organized as follows: Section 2 gives a brief review of the QA task and describe two types of QA systems with different pros and cons. Section 3 presents two MBRAR approaches that can re-rank the answer candidates from single and multiple QA systems respectively. The relationship between our approach and previous work is discussed in Section 4. Section 5 evaluates our methods on large scale questions selected from two domains (Jeopardy! and Web) and shows promising results. Section 6 concludes this paper.

2 Question Answering

2.1 Overview

Formally, given an input question Q, a typical factoid QA system generates answers on the basis of the following three procedures:

(1) *Question Understanding*, which determines the answer type and identifies necessory information contained in Q, such as question focus and lexical answer type (LAT). Such information will be encoded and used by the following procedures.

(2) *Passage Retrieval*, which formulates queries based on Q, and retrieves passages from offline corpus or online search engines (e.g. Google and Bing).

(3) *Answer Extraction*, which first extracts answer candidates from retrieved passages, and then ranks them based on specific ranking models.

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 424–428,
Sofia, Bulgaria, August 4-9 2013.

2.2 Two Types of QA Systems

We present two different QA sysytems, which are distinguished from three aspects: answer typing, answer generation, and answer ranking.

The 1^{st} QA system is denoted as *Type-Dependent* QA engine (**TD-QA**). In answer typing phase, TD-QA assigns the most possible answer type $\hat{T}$ to a given question Q based on:

$$\hat{T} = \underset{T}{argmax}\, P(T|Q)$$

$P(T|Q)$ is a probabilistic answer-typing model that is similar to Pinchak and Lin (2006)'s work. In answer generation phase, TD-QA uses a CRF-based Named Entity Recognizer to detect all named entities contained in retrieved passages with the type $\hat{T}$, and treat them as the answer candidate space $\mathcal{H}(Q)$:

$$\mathcal{H}(Q) = \bigcup_{k} \mathcal{A}_k$$

In answer ranking phase, the decision rule described below is used to rank answer candidate space $\mathcal{H}(Q)$:

$$\begin{aligned} \hat{\mathcal{A}} &= \underset{\mathcal{A}\in\mathcal{H}(Q)}{argmax}\, P(\mathcal{A}|\hat{T}, Q) \\ &= \underset{\mathcal{A}\in\mathcal{H}(Q)}{argmax} \sum_{i} \lambda_i \cdot h_i(\mathcal{A}, \hat{T}, Q) \end{aligned}$$

where $\{h_i(\cdot)\}$ is a set of ranking features that measure the correctness of answer candidates, and $\{\lambda_i\}$ are their corresponding feature weights.

The 2^{ed} QA system is denoted as *Type-Independent* QA engine (**TI-QA**). In answer typing phase, TI-QA assigns top N, instead of the best, answer types $\mathcal{T}_N(Q)$ for each question Q. The probability of each type candidate is maintained as well. In answer generation phase, TI-QA extracts all answer candidates from retrieved passages based on answer types in $\mathcal{T}_N(Q)$, by the same NER used in TD-QA. In answer ranking phase, TI-QA considers the probabilities of different answer types as well:

$$\begin{aligned} \hat{\mathcal{A}} &= \underset{\mathcal{A}\in\mathcal{H}(Q)}{argmax}\, P(\mathcal{A}|Q) \\ &= \underset{\mathcal{A}\in\mathcal{H}(Q)}{argmax} \sum_{T\in\mathcal{T}_N(Q)} P(\mathcal{A}|T, Q) \cdot P(T|Q) \end{aligned}$$

On one hand, TD-QA can achieve relative high ranking precision, as using a unique answer type greatly reduces the size of the candidate list for ranking. However, as the answer-typing model is far from perfect, if prediction errors happen, TD-QA can no longer give correct answers at all.

On the other hand, TI-QA can provide higher answer coverage, as it can extract answer candidates with multiple answer types. However, more answer candidates with different types bring more difficulties to the answer ranking model to rank the correct answer to the top 1 position. So the ranking precision of TI-QA is not as good as TD-QA.

3 MBR-based Answering Re-ranking

3.1 MBRAR for Single QA System

MBR decoding (Bickel and Doksum, 1977) aims to select the hypothesis that minimizes the expected loss in classification. In MBRAR, we replace the loss function with the gain function that measure the correlation between answer candidates. Thus, the objective of the MBRAR approach for single QA system is to find the answer candidate that is most supported by other candidates under QA system's distribution, which can be formally written as:

$$\hat{\mathcal{A}} = \underset{\mathcal{A}\in\mathcal{H}(Q)}{argmax} \sum_{\mathcal{A}_k\in\mathcal{H}(Q)} \mathcal{G}(\mathcal{A}, \mathcal{A}_k) \cdot P(\mathcal{A}_k|\mathcal{H}(Q))$$

$P(\mathcal{A}_k|\mathcal{H}(Q))$ denotes the *hypothesis distribution* estimated on the search space $\mathcal{H}(Q)$ based on the following log-linear formulation:

$$P(\mathcal{A}_k|\mathcal{H}(Q)) = \frac{exp(\beta \cdot P(\mathcal{A}_k|Q))}{\sum_{\mathcal{A}'\in\mathcal{H}} exp(\beta \cdot P(\mathcal{A}'|Q))}$$

$P(\mathcal{A}_k|Q)$ is the posterior probability of the answer candidate $\mathcal{A}_k$ based on QA system's ranking model, β is a scaling factor which controls the distribution $P(\cdot)$ sharp (when $\beta > 1$) or smooth (when $\beta < 1$).

$\mathcal{G}(\mathcal{A}, \mathcal{A}_k)$ is the *gain function* that denotes the degree of how $\mathcal{A}_k$ supports $\mathcal{A}$. This function can be further expanded as a weighted combination of a set of correlation features as: $\sum_j \lambda_j \cdot h_j(\mathcal{A}, \mathcal{A}_k)$. The following correlation features are used in $\mathcal{G}(\cdot)$:

- answer-level n-gram correlation feature:

$$h_{answer}(\mathcal{A}, \mathcal{A}_k) = \sum_{\omega\in\mathcal{A}} \#_{\omega}(\mathcal{A}_k)$$

  where ω denotes an n-gram in $\mathcal{A}$, $\#_{\omega}(\mathcal{A}_k)$ denotes the number of times that ω occurs in $\mathcal{A}_k$.

- passage-level n-gram correlation feature:

$$h_{passage}(\mathcal{A}, \mathcal{A}_k) = \sum_{\omega \in \mathcal{P}_{\mathcal{A}}} \#_{\omega}(\mathcal{P}_{\mathcal{A}_k})$$

  where $\mathcal{P}_{\mathcal{A}}$ denotes passages from which $\mathcal{A}$ are extracted. This feature measures the degree of $\mathcal{A}_k$ supports $\mathcal{A}$ from the context perspective.

- answer-type agreement feature:

$$h_{type}(\mathcal{A}, \mathcal{A}_k) = \delta(T_{\mathcal{A}}, T_{\mathcal{A}_i})$$

  $\delta(T_{\mathcal{A}}, T_{\mathcal{A}_k})$ denotes an indicator function that equals to 1 when the answer types of $\mathcal{A}$ and $\mathcal{A}_k$ are the same, and 0 otherwise.

- answer-length feature that is used to penalize long answer candidates.

- averaged passage-length feature that is used to penalize passages with a long averaged length.

3.2 MBRAR for Multiple QA Systems

Aiming to apply MBRAR to the outputs from N QA systems, we modify MBR components as follows.

First, the hypothesis space $\mathcal{H}_{\mathcal{C}}(Q)$ is built by merging answer candidates of multiple QA systems:

$$\mathcal{H}_{\mathcal{C}}(Q) = \bigcup_{i} \mathcal{H}_i(Q)$$

Second, the hypothesis distribution is defined as a probability distribution over the combined search space of N component QA systems and computed as a weighted sum of component model distributions:

$$P(\mathcal{A}|\mathcal{H}_{\mathcal{C}}(Q)) = \sum_{i=1}^{N} \alpha_i \cdot P(\mathcal{A}|\mathcal{H}_i(Q))$$

where $\alpha_1, ..., \alpha_N$ are coefficients with following constraints holds[1]: $0 \leq \alpha_i \leq 1$ and $\sum_{i=1}^{N} \alpha_i = 1$, $P(\mathcal{A}|\mathcal{H}_i(Q))$ is the posterior probability of $\mathcal{A}$ estimated on the i^{th} QA system's search space $\mathcal{H}_i(Q)$.

Third, the features used in the gain function $\mathcal{G}(\cdot)$ can be grouped into two categories, including:

- *system-independent features*, which includes all features described in Section 3.1 for single system based MBRAR method;

- *system-dependent features*, which measure the correctness of answer candidates based on information provided by multiple QA systems:
  - system indicator feature $h_{sys}(\mathcal{A}, QA_i)$, which equals to 1 when $\mathcal{A}$ is generated by the i^{th} system QA_i, and 0 otherwise;
  - system ranking feature $h_{rank}(\mathcal{A}, QA_i)$, which equals to the reciprocal of the rank position of $\mathcal{A}$ predicted by QA_i. If QA_i fails to generate $\mathcal{A}$, then it equals to 0;
  - ensemble feature $h_{cons}(\mathcal{A})$, which equals to 1 when $\mathcal{A}$ can be generated by all individual QA system, and 0 otherwise.

Thus, the MBRAR for multiple QA systems can be finally formulated as follows:

$$\hat{\mathcal{A}} = \underset{\mathcal{A} \in \mathcal{H}_{\mathcal{C}}(Q)}{argmax} \sum_{\mathcal{A}_i \in \mathcal{H}_{\mathcal{C}}(Q)} \mathcal{G}(\mathcal{A}, \mathcal{A}_i) \cdot P(\mathcal{A}_i|\mathcal{H}_{\mathcal{C}}(Q))$$

where the training process of the weights in the gain function is carried out with Ranking SVM[2] based on the method described in Verberne et al. (2009).

4 Related Work

MBR decoding have been successfully applied to many NLP tasks, e.g. machine translation, parsing, speech recognition and etc. As far as we know, this is the first work that applies MBR principle to QA.

Yaman et al. (2009) proposed a classification based method for QA task that jointly uses multiple 5-W QA systems by selecting one optimal QA system for each question. Comparing to their work, our MBRAR approaches assume few about the question types, and all QA systems contribute in the re-ranking model. Tellez-Valero et al. (2008) presented an answer validation method that helps individual QA systems to automatically detect its own errors based on information from multiple QA systems. Chu-Carroll et al. (2003) presented a multi-level answer resolution algorithm to merge results from the answering agents at the question, passage, and answer levels. Grappy et al.

[1] For simplicity, the coefficients are equally set: $\alpha_i = 1/N$.

[2] We use SVM^{Rank} (Joachims, 2006) that can be founded at www.cs.cornell.edu/people/tj/svm light/svm rank.html/

(2012) proposed to use different score combinations to merge answers from different QA systems. Although all methods mentioned above leverage information provided by multiple QA systems, our work is the first time to explore the usage of MBR principle for the QA task.

5 Experiments

5.1 Data and Metric

Questions from two different domains are used as our evaluation data sets: the first data set includes 10,051 factoid question-answer pairs selected from the Jeopardy! quiz show[3]; while the second data set includes 360 celebrity-asking web questions[4] selected from a commercial search engine, the answers for each question is labeled by human annotators.

The evaluation metric $Succeed@n$ is defined as the number of questions whose correct answers are successfully ranked to the top n answer candidates.

5.2 MBRAR for Single QA System

We first evaluate the effectiveness of our MBRAR for single QA system. Given the N-best answer outputs from each single QA system, together with their ranking scores assigned by the corresponding ranking components, we further perform MBRAR to re-rank them and show resulting numbers on two evaluation data sets in Table 1 and 2 respectively.

Both Table 1 and Table 2 show that, by leveraging our MBRAR method on individual QA systems, the rankings of correct answers are consistently improved on both Jeopardy! and web questions.

Joepardy!	$Succeed@1$	$Succeed@2$	$Succeed@3$
TD-QA	2,289	2,693	2,885
MBRAR	**2,372**	**2,784**	**2,982**
TI-QA	2,527	3,397	3,821
MBRAR	**2,628**	**3,500**	**3,931**

Table 1: Impacts of MBRAR for single QA system on Jeopardy! questions.

We also notice TI-QA performs significantly better than TD-QA on Jeopardy! questions, but worse on web questions. This is due to fact that when the answer type is fixed (PERSON for celebrity-asking questions), TI-QA will generate candidates with wrong answer types, which will definitely deteriorate the ranking accuracy.

[3]http://www.jeopardy.com/

[4]The answers of such questions are person names.

Web	$Succeed@1$	$Succeed@2$	$Succeed@3$
TD-QA	97	128	146
MBRAR	**99**	**130**	**148**
TI-QA	95	122	136
MBRAR	**97**	**126**	**143**

Table 2: Impacts of MBRAR for single QA system on web questions.

5.3 MBRAR for Multiple QA Systems

We then evaluate the effectiveness of our MBRAR for multiple QA systems. The mixture model-based MBRAR method described in Section 3.2 is used to rank the combined answer outputs from TD-QA and TI-QA, with ranking results shown in Table 3 and 4.

From Table 3 and Table 4 we can see that, comparing to the ranking performances of single QA systems TD-QA and TI-QA, MBRAR using two QA systems' outputs shows significant improvements on both Jeopardy! and web questions. Furthermore, comparing to MBRAR on single QA system, MBRAR on multiple QA systems can provide extra gains on both questions sets as well.

Jeopardy!	$Succeed@1$	$Succeed@2$	$Succeed@3$
TD-QA	2,289	2,693	2,885
TI-QA	2,527	3,397	3,821
MBRAR	**2,891**	**3,668**	**4,033**

Table 3: Impacts of MBRAR for multiple QA systems on Jeopardy! questions.

Web	$Succeed@1$	$Succeed@2$	$Succeed@3$
TD-QA	97	128	146
TI-QA	95	122	136
MBRAR	**108**	**137**	**152**

Table 4: Impacts of MBRAR for multiple QA systems on web questions.

6 Conclusions and Future Work

In this paper, we present two MBR-based answer re-ranking approaches for QA. Comparing to previous methods, MBRAR provides a systematic way to re-rank answers from either single or multiple QA systems, without considering their heterogeneous implementations of internal components.

Experiments on questions from two different domains show that, our proposed method can significantly improve the ranking performances. In future, we will add more QA systems into our M-BRAR framework, and design more features for the MBR gain function.

References

P. J. Bickel and K. A. Doksum. 1977. *Mathematical Statistics: Basic Ideas and Selected Topics.* Holden-Day Inc.

Jennifer Chu-Carroll, Krzysztof Czuba, John Prager, and Abraham Ittycheriah. 2003. *In Question Answering, Two Heads Are Better Than One*. In proceeding of HLT-NAACL.

Vaibhava Goel and William Byrne. 2000. *Minimum bayes-risk automatic speech recognition*, Computer Speech and Language.

Arnaud Grappy, Brigitte Grau, and Sophie Rosset. 2012. *Methods Combination and ML-based Re-ranking of Multiple Hypothesis for Question-Answering Systems*, In proceeding of EACL.

Thorsten Joachims. 2006. *Training Linear SVMs in Linear Time*, In proceeding of KDD.

Shankar Kumar and William Byrne. 2004. *Minimum Bayes-Risk Decoding for Statisti-cal Machine Translation*. In proceeding of HLT-NAACL.

Christopher Pinchak and Dekang Lin. 2006. *A Probabilistic Answer Type Model*. In proceeding of EACL.

Ivan Titov and James Henderson. 2006. *Bayes Risk Minimization in Natural Language Parsing*. Technical report.

Alberto Tellez-Valero, Manuel Montes-y-Gomez, Luis Villasenor-Pineda, and Anselmo Penas. 2008. *Improving Question Answering by Combining Multiple Systems via Answer Validation*. In proceeding of CICLing.

Suzan Verberne, Clst Ru Nijmegen, Hans Van Halteren, Clst Ru Nijmegen, Daphne Theijssen, Ru Nijmegen, Stephan Raaijmakers, Lou Boves, and Clst Ru Nijmegen. 2009. *Learning to rank qa data. evaluating machine learning techniques for ranking answers to why-questions*. In proceeding of SIGIR workshop.

Sibel Yaman, Dilek Hakkani-Tur, Gokhan Tur, Ralph Grishman, Mary Harper, Kathleen R. McKeown, Adam Meyers, Kartavya Sharma. 2009. *Classification-Based Strategies for Combining Multiple 5-W Question Answering Systems*. In proceeding of INTERSPEECH.