ECAI 2008
M. Ghallab et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-891-5-839

Answering Definition Question: Ranking for Top-k

Chao Shen and **Xipeng Qiu** and **Xuanjing Huang** and **Lide Wu** [1]

Abstract. As an important form of complex questions, definition question attracts much attention from QA researchers. For many of the definition question answering systems, it is a core step to rank the candidate answer sentences, so that the top-k in the ranked list can be extracted. We integrate these evidences as features into a whole framework, and propose a novel method to learning weights of these features to rank the candidate answer sentences.

1 Introduction

Definition question answering[10], as an important form of complex question answering is attracting more attention recently. The definition question can be interpreted as "Tell me interesting things about X". Here "X" is usually called "target".

Most definition question answering systems have the pipeline structure:

Step-1 Extracting the candidate answer sentences from the corpus.
Step-2 Ranking the candidate answer sentences.
Step-3 Removing redundant answer sentences.

Step-1 is the IR on the sentence or sub-sentence level. For a target, we can get a list of sentences through this step. Step-2 is the core step, which ranks the output of Step-1. Many researches on definition question focus on this step and various methods have been developed. Some simple methods such as checking the overlap of words between two sentences in the answer are often used in the step-3.

To answer definition questions, pattern based methods [3] and centroid vector based methods [1, 5] are popular in ranking the answer sentences. And various resources including lexico-syntactic patterns and external resources such as Google, Wikipedia, encyclopedia, have been used as evidences to judge whether a sentence is a definition sentence about a target. However, in previous systems, if multiple resources have been used, the importance of each resource in the definition question answering system is fixed manually. Since different patterns and centroid vector may play different roles, there should be a way to automatically identify their weights.

Our work propose a learning method which 1) gives the optimal top-k sentences instead of the optimal ranking of the whole list and 2) explicitly slackens the condition that definition sentences should be ranked ahead of the other. Using such learning method for ranking, we integrate evidences for sentence be to definition as features into a whole framework and achieve a better result.

2 Learning to Rank for Top-k

In this section, we will introduce how weights of resources is learned. Specifically, we use modified version of a online learning algorithm MIRA [2] for the task of sentence ranking in definition question answering. In training, a set of targets $X = \{x^1, x^2, \ldots, x^T\}$ is given. Each target x^t is associated with a set of nuggets sentences $y^t = \{y_1^t, y_2^t, \ldots, y_{n^t}^t\}$, where y_j^t denotes the j-th sentence and n^t denotes the sizes of y^t. Meanwhile, each target also associated a list of sentences, $s^t = \{s_1^t, s_2^t, \ldots, s_{m^t}^t\}$, which are the output of the first step of the pipeline system, and to be ranked. From s^t, we will select k sentences as the input of the step 3 module or directly as the answer of the target. An arbitary subset of s^t with size k is denoted as $s^t(k)$. To evaluate these sets of sentences, we defined $score(x^t, s^t(k)) = w * \Psi(x^t, s^t(k))$, where $\Psi(x^t, s^t(k))$ is the feature vector for the target and its k-sentences pair $< x^t, s^t(k) >$ and $\hat{y^t} = \arg\max_{s^t(k)} score(x^t, s^t(k))$ will be extracted. We learn w with the goal that as many elements in $\hat{y^t}$ are in y^t as possible.

If we assume each sentence is independent with others, the feature of the <target,k-sentences> pair can be defined as $\Psi(x^t, s^t(k)) = \sum_{j=1}^{k} \psi(x^t, s_j^t)$, where $\psi(x^t, s_j^t)$ is the feature vector for the target-setence pair $< x^t, s_j^t >$ and we can get

$$score(x^t, s^t(k)) = \sum_{j=1}^{k} score(x^t, s_j^t)$$

where $score(x^t, s_j^t) = w * \psi(x^t, s_j^t)$. Thus $\hat{y^t}$ is the top k sentences in the decreasingly ranked list of s^t by $score(x^t, s_j^t)$.

Algorithm 1 Modified Version of Online MIRA

Training Data: $\Gamma = \{(x^t, y^t)_{t=1}^T$
1: $w_0 = 0$; $v = 0$; $i = 0$
2: **for** $n : 1 \ldots N$ **do**
3: **for** $t : 1 \ldots T$ **do**
4: $\min ||w_{i+1} - w_i||$
5: s.t. $score(x^t, s_i^t) - score(x^t, s_j^t) \geq 1$
6: $\forall s_j^t \in Q, \forall s_i^t \in y^{*t} = (y^t \setminus Q) \cup P$
7: $v = v + w_{(i+1)}$
8: $i = i + 1$
9: **end for**
10: **end for**
11: $w = v/(N * T)$

MIRA is first proposed for multi-classification. In [8, 7], it was successfully used for structure-learning. The difference between the MIRA in [8] and the version of ours Algorithm 1 is the contraints (5,6th line of Algorithm 1) used to update the w_i .

To circumvent the problems of ranking for definition question answering mentioned in the Section 1, we first introduce y^{*t}, through adding nugget sentences in $s^t \setminus \hat{y^t}$ to $\hat{y^t}$ and excluding non-nugget sentences from $\hat{y^t}$, and take it as a slackened supervisor of the learning.

[1] Fudan University, China, email: {shenchao,xpqiu,xjhuang,ldwu}@fudan .edu.cn

We define $\theta_0^t = \min\{|\hat{y^t} \setminus y^t|, |(s_t \setminus \hat{y^t}) \cap y^t|\}$, i.e. the minimal number of non-nugget sentence in top-k and non-nugget sentences out of top-k. In the iteration of updating w with the input of (x_t, y_t), we build y^{*t} by inserting $\theta^t = \min\{\theta_0^t, \theta\}$ nugget sentences, P, out of top-k into the top-k sentences and excluding the same number of non-nugget sentences, Q. Then $y^{*t} = (y^t \setminus Q) \cup P$ is a better answer which contains more θ nugget sentences, if possible, compared with $\hat{y^t}$. and P and Q is defined as following:

$$P: \text{ the top-}\theta \text{ nugget sentences of } s^t \setminus \hat{y^t}$$
$$Q: \text{ the bottom-}\theta \text{ non-nugget sentences in } \hat{y^t}$$

3 Experiments

We do two experiments on 65 TREC 2004 targets, 75 TREC 2005 targets and 75 TREC 2006 targets to validate our method. Same module of sentence extraction in [9] is used to extracting the candidate answer sentences from the corpus and no removing redundancy module is used. The features used in the paper is also same in [9], including 4 based on language model, 1 about document retrivel, and several based on syntaical patterns.

In order to building training corpus, we collect the judgement of TREC to all the submitted answers from participants. If a [string, docid] pair is judged covering certain nugget of a target x_t, we extract the original sentence from AQUAINT according to the [string, docid] pair, and add it to the set y_t for target x_t.

3.1 Ranking Comparison

To show the effectiveness of our ranking method, we compare our result with those of the following methods.

RankSVM RankSVM is used to rank definition sentences [11]. As same as in [11], we only use linear kernel.

Han-Model If we fix the weights for 4 features based on language models, we can take our system as a simple version of the statistical model proposed by [4].

Exact-Answer In our proposed method, we do not ask all nugget sentences ranked higher than non-nugget sentences. In this baseline, we construct stricter constraints, all nuggets sentences of a target should be ranked higher than the current non-nuggets sentences in top-k.

$$\begin{aligned} &\text{s.t. } s(x^t, y_i^t) - s(x^t, s_j^t) \geq 1 \\ &\forall s_j^t \in \hat{y^t} \setminus y^t, \forall y_i^t \in y^t \end{aligned} \quad (1)$$

Comparison is on the TREC 2006 targets and TREC 2005 targets are used for training. This is because target set of TREC 2005 and 2006 both include PERSON, ORGANIZATION, THING, EVENT, but TREC 2004 does not contain EVENT targets. θ is decided by 5-fold cross validation on TREC 2005 targets.

Table 1 shows the F3-score for each methed. Though RankSVM and Exact-Answer use more features, they still fail to outperform Han-Model. This implies the importance of the ranking method: If the weights of the features cannot be decided properly, the extra features will not help improve the performance. We can see our method has advandage, especially when k is relative small.

3.2 Comperison with Other Systems

In [5], two state of the art systems, Soft Pattern model(SP) and Human Interests Model(HIM) are evaluated on the TREC 2005 targets with a automatical evaluation tool Pourpre v1.0c [6]. [5] gave the result of their experiment on the TREC 2005 as test data and TREC 2004 as training data. As same as the setting of [5], we select the top 12 highest ranked sentences ($k = 12$) as answers. According to the analysis of the parameter θ, we let $\theta = 2$. From Table 2, we can see our method clearly outperforms SP, and has a comparable result with HIM.

Table 1. Comparison in terms of ranking on the TREC 2006 question set

	F3			
k	Our Method	RankSVM	Han-Model	Exact-Answer
10	**0.2401**	0.1697	0.2282	0.1842
15	**0.2725**	0.2068	0.2382	0.2100
20	**0.2859**	0.2186	0.2592	0.2737
25	**0.2801**	0.2225	0.2610	0.2643
30	**0.2579**	0.1944	0.2557	0.2449
35	0.2338	0.1916	**0.2502**	0.2153

Table 2. Performance on TREC 2005 Question Set

System	F3-Score
Soft-Pattern (SP)	0.2872
Human Interest Model (HIM)	0.3031
Our Method	0.3095

4 Conclusion

In this paper, we integrate multiple resources to rank candidate answer sentences for definition question answering. Specifically, we have proposed a method of learning for ranking to do such task. Instead hoping that all definition sentences are at the top of the list of candidate answer sentences, we use a slack parameter θ to let the top-k sentences involve as many definition sentences as possible.

Experimental results indicate that our proposed method performed better than the several other methods to rank used in the definition question answering. And our multiple resources integrated system has a comparable result to state of the art system.

REFERENCES

[1] Y. Chen, M. Zhou, and S. Wang, 'Reranking answers for definitional QA using language modeling', *Proc. of ACL*, (2006).
[2] K. Crammer and Y. Singer, 'Ultraconservative online algorithms for multiclass problems', *Journal of Machine Learning Research*, **3**, 951–991, (2003).
[3] H. Cui and M.Y. Kan, 'Generic soft pattern models for definitional question answering', *Proc. of ACL*, (2005).
[4] K.S. Han, Y.I. Song, and H.C. Rim, 'Probabilistic model for definitional question answering', *Proc. of SIGIR*, (2006).
[5] K.W. Kor and T.S. Chua, 'Interesting nuggets and their impact on definitional question answering', *Proc. of SIGIR*, (2007).
[6] J. Lin and D. Demner-Fushman, 'Automatically evaluating answers to definition questions', *Proc. of HLT-EMNLP*, (2005).
[7] R. McDonald, 'Discriminative Sentence Compression with Soft Syntactic Evidence', *Proc. of EACL*, (2006).
[8] R. McDonald, K. Crammer, and F. Pereira, 'Online Large-Margin Training of Dependency Parsers', *Proc. of ACL*, (2005).
[9] X. Qiu, B. Li, C. Shen, L. Wu, X. Huang, and Y. Zhou, 'FDUQA on TREC2007 QA Track', *Proc. of TREC*, (2007).
[10] E.M. Voorhees, 'Overview of the TREC 2003 Question Answering Track', *Proc. of TREC*, (2003).
[11] J. Xu, Y. Cao, H. Li, and M. Zhao, 'Ranking definitions with supervised learning methods', *Proc. of WWW*, (2005).