Information Processing and Management 47 (2011) 825–842

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Linguistic kernels for answer re-ranking in question answering systems

Alessandro Moschitti, Silvia Quarteroni *

University of Trento, Via Sommarive 14, 38050 Povo, Trento, Italy

ARTICLE INFO

Article history:
Received 29 March 2009
Received in revised form 2 June 2010
Accepted 7 June 2010
Available online 20 July 2010

Keywords:
Question answering
Information Retrieval
Kernel methods
Predicate argument structures

ABSTRACT

Answer selection is the most complex phase of a question answering (QA) system. To solve this task, typical approaches use unsupervised methods such as computing the similarity between query and answer, optionally exploiting advanced syntactic, semantic or logic representations.

In this paper, we study supervised discriminative models that learn to select (rank) answers using examples of question and answer pairs. The pair representation is implicitly provided by kernel combinations applied to each of its members. To reduce the burden of large amounts of manual annotation, we represent question and answer pairs by means of powerful generalization methods, exploiting the application of structural kernels to syntactic/semantic structures.

We experiment with support vector machines and string kernels, syntactic and shallow semantic tree kernels applied to part-of-speech tag sequences, syntactic parse trees and predicate argument structures on two datasets which we have compiled and made available. Our results on classification of correct and incorrect pairs show that our best model improves the bag-of-words model by 63% on a TREC dataset. Moreover, such a binary classifier, used as a re-ranker, improves the mean reciprocal rank of our baseline QA system by 13%.

These findings demonstrate that our method automatically selects an appropriate representation of question–answer relations.

1. Introduction

Automatic question answering (QA) systems return concise answers – i.e. sentences or phrases – to questions in natural language. On one hand, QA is interesting from an Information Retrieval (IR) viewpoint as it studies means to satisfy the user's information needs; on the other, the high linguistic complexity of QA systems suggests a need for more advanced natural language techniques, that have been shown to be of limited use for more basic IR tasks, e.g. document retrieval (Allan, 2000).

As a matter of fact, the main source of complexity in QA lies in the question processing and answer extraction steps rather than in document retrieval, a step usually conducted using off-the-shelf IR modules (Chen, Zhou, & Wang, 2006; Collins-Thompson, Callan, Terra, & Clarke, 2004).

In question processing, useful information is gathered from the question to create a query; the latter is submitted to the document retrieval module that provides the set of the most relevant documents. The latter are used by the answer extractor to provide a ranked list of candidate answers. In the answer extraction phase, unsupervised methods are usually applied: a similarity between query and answer (such that higher similarity results in higher rank), is computed using simple *bag-of-words* (BOW) models or more advanced syntactic, semantic or logic representations, e.g. (Yang & Chua, 2003; Hovy,

* Corresponding author.
E-mail addresses: alessandro.moschitti@disi.unitn.it (A. Moschitti), silvia.quarteroni@disi.unitn.it (S. Quarteroni).

0306-4573/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2010.06.002

Hermjakob, & Lin, 2001). More recently, shallow semantic information has been successfully exploited for such an approach in terms of predicate argument structures (PASs) (Shen & Lapata, 2007).

In contrast, supervised machine learning methods that learn to rank answers from examples of question and answer pairs (Sasaki, 2005; Suzuki, Sasaki, & Maeda, 2002) rarely use representation more complex than BOW. This is a major drawback, since different questions need different training data, and the only solution to overcome the burden of manual annotation is to reduce it by generalizing such data in terms of syntactic/semantic structures. In previous work, this consideration led us to defining supervised approaches to answer extraction using syntactic and shallow semantic structures. In particular, we proposed two tree kernel functions, named shallow semantic tree kernel (SSTK) (Moschitti, Quarteroni, Basili, & Manandhar, 2007) and partial tree kernel (PTK) (Moschitti & Quarteroni, 2008), that exploit PASs in PropBank[1] format for automatic answering to description questions. The use of shallow semantics appears to be especially relevant in the case of non-factoid questions, such as those requiring definitions, where the answer can be a whole sentence or paragraph containing only one question word.

In this paper, we present a thorough study on the above ideas by focusing on the use of kernel functions to exploit syntactic/semantic structures for relational learning from questions and answers. We start our study from simple linguistic levels and gradually introduce more and more advanced language technology. In more detail, we: (i) model sequence kernels for words and part-of-speech tags that capture basic lexical semantics and syntactic information, (ii) apply tree kernels to encode deeper syntactic information and more structured shallow semantics and (iii) analyze the proposed shallow semantic kernels in terms of both accuracy and efficiency. Finally, we carry out comparative experiments between the different linguistic/kernel models on question/answer classification by measuring the impact of the corresponding classifiers on answer re-ranking.

It is worth noting that, since finding a suitable question answering corpus for our study was difficult,[2] we designed and made available two different corpora, named WEB-QA and TREC-QA. Their main characteristic is that they relate to description questions from TREC 2001 (Voorhees, 2001), whose answers, retrieved from Web and TREC data, respectively, were manually annotated by our group.

The extensive experiments carried out on such corpora show that the generalization ability of kernel functions, successfully used in previous approaches (Collins & Duffy, 2002; Kudo & Matsumoto, 2003; Cumby & Roth, 2003; Culotta & Sorensen, 2004; Kudo, Suzuki, & Isozaki, 2005; Toutanova, Markova, & Manning, 2004; Kazama & Torisawa, 2005; Zhang, Zhang, & Su, 2006), is essential. Indeed, a unique result of our approach is that kernels applied to pairs of questions and answers are effective for automatically learning their relations. This is a further step in automation with respect to previous work such as (Echihabi & Marcu, 2003), that required human effort and intuition to design a structural representation of question–answer pairs and use the latter to extract an effective feature vector representation.[3]

Our main findings are that (i) kernels based on PAS, POS-tag sequences and syntactic parse trees improve on the BOW approach on both datasets: on TREC-QA, the improvement is high (about 63% in F1 score), making its application worthwhile; (ii) PTK for processing PASs is more efficient and effective than SSTK and can be practically used in answer re-ranking systems; and (iii) our best question/answer classifier, used as a re-ranker, improves the mean reciprocal rank (MRR) of our QA basic system by 13%, confirming its promising applicability. Such improvement is much larger on WEB-QA.

In the remainder of this paper, Section 2 presents our use of kernel functions for structural information and Section 3 introduces the data representations we use for question and answer pairs. Section 4 reports on our experiments with different learning models and representations. Finally, Section 5 discusses our approach with respect to related work and our final conclusions are drawn in Section 6.

2. Kernel methods for structured data

Kernel methods refer to a large class of learning algorithms based on inner product vector spaces, among which support vector machines (SVMs) are well-known algorithms. The main idea behind SVMs is to learn a hyperplane $H(\vec{x}) = \vec{w} \cdot \vec{x} + b = 0$, where $\vec{x}$ is the representation of a classifying object o as a feature vector, while $\vec{w} \in \Re^n$ (indicating that $\vec{w}$ belongs to a vector space of n dimensions built on real numbers) and $b \in \Re$ are parameters learnt from training examples by applying the *Structural Risk Minimization principle* (Vapnik, 1995). Object o is mapped into $\vec{x}$ via a feature function $\phi : \mathcal{O} \to \Re^n$, where $\mathcal{O}$ is the set of objects; o is categorized in the target class only if $H(\vec{x}) \geqslant 0$.

By exploiting the "kernel trick", the decision hyperplane can be rewritten as:

$$H(\vec{x}) = \left(\sum_{i=1,\ldots,l} y_i \alpha_i \vec{x}_i\right) \cdot \vec{x} + b = \sum_{i=1,\ldots,l} y_i \alpha_i \vec{x}_i \cdot \vec{x} + b = \sum_{i=1,\ldots,l} y_i \alpha_i \phi(o_i) \cdot \phi(o) + b,$$

where y_i is equal to 1 for positive examples and to -1 for negative examples, $\alpha_i \in \Re$ (with $\alpha_i \geqslant 0$, $o_i\ \forall\ i \in \{1,\ldots,l\}$) are the training instances and the product $K(o_i, o) = \langle \phi(o_i) \cdot \phi(o) \rangle$ is the kernel function associated with the mapping ϕ.

[1] www.cis.upenn.edu/ace.

[2] For supervised approaches we could use neither the Japanese corpus used in Sasaki (2005), Suzuki et al. (2002) nor the corpus used in Echihabi and Marcu (2003), since they are not publicly available.

[3] Machine translation techniques were applied to make this task easier.