Kernels on Linguistic Structures for Answer Extraction

Alessandro Moschitti and **Silvia Quarteroni**
DISI, University of Trento
Via Sommarive 14
38100 POVO (TN) - Italy
{moschitti,silviaq}@disi.unitn.it

Abstract

Natural Language Processing (NLP) for Information Retrieval has always been an interesting and challenging research area. Despite the high expectations, most of the results indicate that successfully using NLP is very complex. In this paper, we show how Support Vector Machines along with kernel functions can effectively represent syntax and semantics. Our experiments on question/answer classification show that the above models highly improve on bag-of-words on a TREC dataset.

1 Introduction

Question Answering (QA) is an IR task where the major complexity resides in question processing and answer extraction (Chen et al., 2006; Collins-Thompson et al., 2004) rather than document retrieval (a step usually carried out by off-the shelf IR engines). In question processing, useful information is gathered from the question and a query is created. This is submitted to an IR module, which provides a ranked list of relevant documents. From these, the QA system extracts one or more candidate answers, which can then be re-ranked following various criteria. Although typical methods are based exclusively on word similarity between query and answer, recent work, e.g. (Shen and Lapata, 2007) has shown that shallow semantic information in the form of predicate argument structures (PASs) improves the automatic detection of correct answers to a target question. In (Moschitti et al., 2007), we proposed the Shallow Semantic Tree Kernel (SSTK) designed to encode PASs[1] in SVMs.

In this paper, similarly to our previous approach, we design an SVM-based answer extractor, that selects the correct answers from those provided by a basic QA system by applying tree kernel technology. However, we also provide: (i) a new kernel to process PASs based on the partial tree kernel algorithm (PAS-PTK), which is highly more efficient and more accurate than the SSTK and (ii) a new kernel called Part of Speech sequence kernel (POSSK), which proves very accurate to represent shallow syntactic information in the learning algorithm.

To experiment with our models, we built two different corpora, WEB-QA and TREC-QA by using the description questions from TREC 2001 (Voorhees, 2001) and annotating the answers retrieved from Web resp. TREC data (available at disi.unitn.it/~silviaq). Comparative experiments with re-ranking models of increasing complexity show that: (a) PAS-PTK is far more efficient and effective than SSTK, (b) POSSK provides a remarkable further improvement on previous models. Finally, our experiments on the TREC-QA dataset, un-biased by the presence of typical Web phrasings, show that BOW is inadequate to learn relations between questions and answers. This is the reason why our kernels on linguistic structures improve it by 63%, which is a remarkable result for an IR task (Allan, 2000).

2 Kernels for Q/A Classification

The design of an answer extractor basically depends on the design of a classifier that decides if an answer correctly responds to the target question. We design a classifier based on SVMs and different kernels applied to several forms of question and answer

[1] in PropBank format, (www.cis.upenn.edu/~ace).

Proceedings of ACL-08: HLT, Short Papers (Companion Volume), pages 113–116,
Columbus, Ohio, USA, June 2008. ©2008 Association for Computational Linguistics

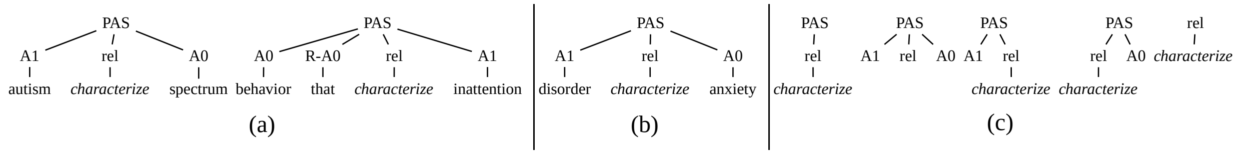

Figure 1: Compact PAS-PTK structures of s_1 (a) and s_2 (b) and some fragments they have in common as produced by the PTK (c). Arguments are replaced with their most important word (or semantic head) to reduce data sparseness.

representations:
(1) linear kernels on the bag-of-words (BOW) or bag-of-POS-tags (POS) features,
(2) the String Kernel (SK) (Shawe-Taylor and Cristianini, 2004) on word sequences (WSK) and POS-tag sequences (POSSK),
(3) the Syntactic Tree Kernel (STK) (Collins and Duffy, 2002) on syntactic parse trees (PTs),
(4) the Shallow Semantic Tree Kernel (SSTK) (Moschitti et al., 2007) and the Partial Tree Kernel (PTK) (Moschitti, 2006) on PASs.

In particular, POS-tag sequences and PAS trees used with SK and PTK yield to two innovative kernels, i.e. POSSK and PAS-PTK[2]. In the next sections, we describe in more detail the data structures on which we applied the above kernels.

2.1 Syntactic Structures

The POSSK is obtained by applying the String Kernel on the sequence of POS-tags of a question or a answer. For example, given sentence s_0: *What is autism?*, the associated POS sequence is *WP AUX NN ?* and some of the substrings extracted by POSSK are *WP NN* or *WP AUX*. A more complete structure is the full parse tree (PT) of the sentence, that constitutes the input of the STK. For instance, the STK accepts the syntactic parse: *(SBARQ (WHNP (WP What))(SQ (VP (AUX is)(NP (NN autism))))(. ?))*.

2.2 Semantic Structures

The intuition behind our semantic representation is the idea that when we ignore the answer to a definition question we check whether such answer is formulated as a "typical" definition and whether answers defining similar concepts are expressed in a similar way.

To take advantage of semantic representations, we work with two types of semantic structures; first, the Word Sequence Kernel applied to both question and answer; given s_0, sample substrings are: *What is autism*, *What is*, *What autism*, *is autism*, etc. Then, two PAS-based trees: Shallow Semantic Trees for SSTK and Shallow Semantic Trees for PTK, both based on PropBank structures (Kingsbury and Palmer, 2002) are automatically generated by our SRL system (Moschitti et al., 2005). As an example, let us consider an automatically annotated sentence from our TREC-QA corpus:
s_1: *[$_{A1}$ Autism] is [$_{rel}$ characterized] [$_{A0}$ by a broad spectrum of behavior] [$_{R-A0}$ that] [$_{rel}$includes] [$_{A1}$ extreme inattention to surroundings and hypersensitivity to sound and other stimuli].*
Such annotation can be used to design a shallow semantic representation that can be matched against other semantically similar sentences, e.g.
s_2: *[$_{A1}$ Panic disorder] is [$_{rel}$ characterized] [$_{A0}$ by unrealistic or excessive anxiety].*
It can be observed here that, although autism is a different disease from panic disorder, the structure of both definitions and the latent semantics they contain (inherent to behavior, disorder, anxiety) are similar. So for instance, s_2 appears as a definition even to someone who only knows what the definition of autism looks like.

The above annotation can be compactly represented by predicate argument structure trees (PASs) such as those in Figure 1. Here, we can notice that the semantic similarity between sentences is explicitly visible in terms of common fragments extracted by the PTK from their respective PASs. Instead, the similar PAS-SSTK representation in (Moschitti et al., 2007) does not take argument order into account, thus it fails to capture the linguistic rationale expressed above. Moreover, it is much heavier, causing large memory occupancy and, as shown by our experiments, much longer processing time.

[2] For example, let PTK$(t_1, t_2) = \phi(t_1) \cdot \phi(t_2)$, where t_1 and t_2 are two syntactic parse trees. If we map t_1 and t_2 into two new shallow semantic trees s_1 and s_2 with a mapping $\phi_M(\cdot)$, we obtain: PTK$(s_1, s_2) = \phi(s_1) \cdot \phi(s_2) = \phi(\phi_M(t_1)) \cdot \phi(\phi_M(t_2)) = \phi'(t_1) \cdot \phi'(t_2)$=PAS-PTK$(t_1, t_2)$, which is a noticeably different kernel induced by the mapping $\phi' = \phi \circ \phi_M$.

3 Experiments

In our experiments we show that (a) the PAS-PTK shallow semantic tree kernel is more efficient and effective than the SSTK proposed in (Moschitti et al., 2007), and (b) our POSSK jointly used with PAS-PTK and STK greatly improves on BOW.

3.1 Experimental Setup

In our experiments, we implemented the BOW and POS kernels, WSK, POSSK, STK (on syntactic PTs derived automatically with Charniak's parser), SSTK and PTK (on PASs derived automatically with our SRL system) as well as their combinations in SVM-light-TK[3]. Since answers often contain more than one PAS (see Figure 1), we sum PTK (or SSTK) applied to all pairs $P_1 \times P_2$, P_1 and P_2 being the sets of PASs of the first two answers.

The experimental datasets were created by submitting the 138 TREC 2001 test questions labeled as "description" in (Li and Roth, 2002) to our basic QA system, YourQA (Quarteroni and Manandhar, 2008) and by gathering the top 20 answer paragraphs.

YourQA was run on two sources: Web documents by exploiting Google (`code.google.com/apis/`) and the AQUAINT data used for TREC'07 (`trec.nist.gov/data/qa`) by exploiting Lucene (`lucene.apache.org`), yielding two different corpora: WEB-QA and TREC-QA. Each sentence of the returned paragraphs was manually evaluated based on whether it contained a correct answer to the corresponding question. To simplify our task, we isolated for each paragraph the sentence with the maximal judgment (such as s_1 and s_2 in Sec. 2.2) and labeled it as positive if it answered the question either concisely or with noise, negative otherwise. The resulting WEB-QA corpus contains 1309 sentences, 416 of which positive; the TREC-QA corpus contains 2256 sentences, 261 of which positive.

3.2 Results

In a first experiment, we compared the learning and classification efficiency of SVMs on PASs by applying either solely PAS-SSTK or solely PAS-PTK on the WEB-QA and TREC-QA sets. We divided the training data in 9 bins of increasing size (with a step of 200) and measured the training and test time[4] for each bin. Figure 2 shows that in both the test and training phases, PTK is much faster than SSTK. In training, PTK is 40 times faster, enabling the experimentation of SVMs with large datasets. This difference is due to the combination of our lighter semantic structures and the PTK's ability to extract from these at least the same information that SSTK derives from much larger structures.

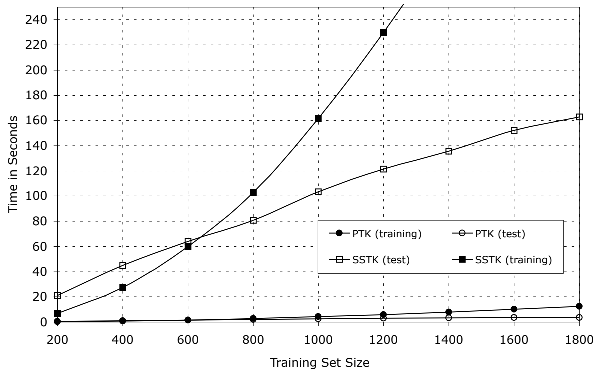

Figure 2: Efficiency of PTK and SSTK

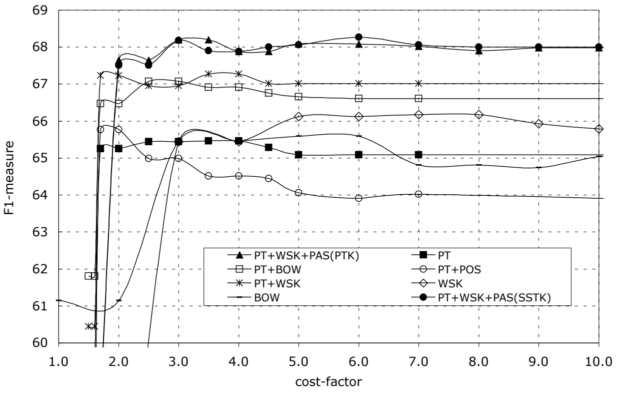

Figure 3: Impact of different kernels on WEB-QA

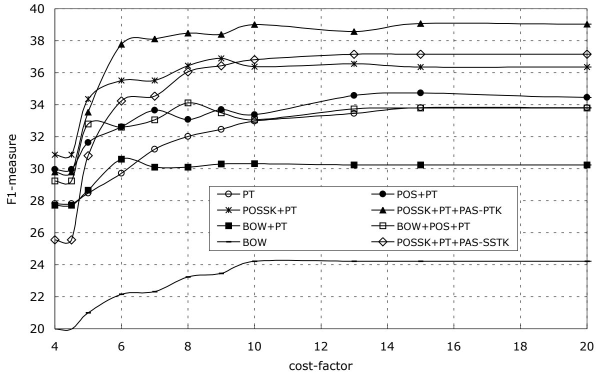

Figure 4: Impact of different kernels on TREC-QA

Further interesting experiments regard the accu-

[3] Toolkit available at `dit.unitn.it/moschitti/`, based on SVM-light (Joachims, 1999)

[4] Processing time in seconds of a Mac-Book Pro 2.4 Ghz.

racy tests of different kernels and some of their most promising combinations. As a kernel operator, we applied the sum between kernels[5] that yields the joint feature space of the individual kernels (Shawe-Taylor and Cristianini, 2004).

Figure 3 shows the F1-plots of several kernels according to different cost-factor values (i.e. different Precision/Recall rates). Each F1 value is the average of 5 fold cross-validation. We note that (a) BOW achieves very high accuracy, comparable to the one produced by PT; (b) the BOW+PT combination improves on both single models; (c) WSK improves on BOW and it is enhanced by WSK+PT, demonstrating that word sequences and PTs are very relevant for this task; (d) both PAS-SSTK and PAS-PTK improve on previous models yielding the highest result.

The high accuracy of BOW is surprising as support vectors are compared with test examples which are in general different (there are no questions shared between training and test set). The explanation resides in the fact that WEB-QA contains common BOW patterns due to typical Web phrasings, e.g. `Learn more about X`, that facilitate the detection of incorrect answers.

Hence, to have un-biased results, we experimented with the TREC corpus which is cleaner from a linguistic viewpoint and also more complex from a QA perspective. A comparative analysis of Figure 4 suggests that: (a) the F1 of all models is much lower than for the WEB-QA dataset; (b) BOW denotes the lowest accuracy; (c) POS combined with PT improves on PT; (d) POSSK+PT improves on POS+PT; (f) finally, PAS adds further information as the best model is POSSK+PT+PAS-PTK (or PAS-SSTK).

4 Conclusions

With respect to our previous findings, experimenting with TREC-QA allowed us to show that BOW is not relevant to learn re-ranking functions from examples; indeed, while it is useful to establish an initial ranking by measuring the similarity between question and answer, BOW is almost irrelevant to grasp typical rules that suggest if a description is valid or not. Moreover, using the new POSSK and PAS-PTK kernels provides an improvement of 5 absolute percent points wrt our previous work.

Finally, error analysis revealed that PAS-PTK can provide patterns like `A1(X) R-A1(that) rel(result) A1(Y)` and `A1(X) rel(characterize) A0(Y)`, where `X` and `Y` need not necessarily be matched.

Acknowledgments

This work was partly supported by the FP6 IST LUNA project (contract No. 33549) and by the European Commission Marie Curie Excellence Grant for the ADAMACH project (contract No. 022593).

References

J. Allan. 2000. Natural language processing for information retrieval. In *Proceedings of NAACL/ANLP (tutorial notes)*.

Y. Chen, M. Zhou, and S. Wang. 2006. Reranking answers from definitional QA using language models. In *ACL'06*.

M. Collins and N. Duffy. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *ACL'02*.

K. Collins-Thompson, J. Callan, E. Terra, and C. L.A. Clarke. 2004. The effect of document retrieval quality on factoid QA performance. In *SIGIR'04*.

T. Joachims. 1999. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*.

P. Kingsbury and M. Palmer. 2002. From Treebank to PropBank. In *LREC'02*.

X. Li and D. Roth. 2002. Learning question classifiers. In *ACL'02*.

A. Moschitti, B. Coppola, A. Giuglea, and R. Basili. 2005. Hierarchical semantic role labeling. In *CoNLL 2005 shared task*.

A. Moschitti, S. Quarteroni, R. Basili, and S. Manandhar. 2007. Exploiting syntactic and shallow semantic kernels for question/answer classification. In *ACL'07*.

A. Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *ECML'06*.

S. Quarteroni and S. Manandhar. 2008. Designing an interactive open domain question answering system. *Journ. of Nat. Lang. Eng. (in press)*.

J. Shawe-Taylor and N. Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

D. Shen and M. Lapata. 2007. Using semantic roles to improve question answering. In *EMNLP-CoNLL*.

E. M. Voorhees. 2001. Overview of the TREC 2001 Question Answering Track. In *TREC'01*.

[5] All adding kernels are normalized to have a similarity score between 0 and 1, i.e. $K'(X_1, X_2) = \frac{K(X_1,X_2)}{\sqrt{K(X_1,X_1) \times K(X_2,X_2)}}$.