Learning to answer questions

Ana Cristina Mendes
L^{2}F INESC-ID Lisboa
Instituto Superior Tecnico

Luísa Coheur
L^{2}F INESC-ID Lisboa
Instituto Superior Tecnico

Sérgio Curto
L^{2}F INESC-ID Lisboa

Abstract

We present an open-domain Question-Answering system that learns to answer questions based on successful past interactions. We follow a pattern-based approach to Answer Extraction, where (lexico-syntactic) patterns that relate a question to its answer are automatically learned and used to answer future questions. Results show that our approach contributes to the system's best performance when it is conjugated with typical Answer Extraction strategies. Moreover, it allows the system to learn with the answered questions and to rectify wrong or unsolved past questions.

1 Introduction

Question-Answering (QA) systems aim at delivering correct answers to questions posed in natural language. In this paper, we present our approach to open-domain QA, motivated by the hypothesis that if a system is given feedback about the correctness of its answers, it can learn to answer new questions.

Our approach relies on iteratively learning to answer new questions based on previously answered questions. In the basis of this work, there is a pattern-based strategy to Answer Extraction (AE), with (lexico-syntactic) patterns learned from pairs composed of natural language questions and their answers. Thus, when a new question is posed to the system, the applicable patterns are unified with the sentences where the new answers might lie. If the unification succeeds, possible answers are extracted; if not, the unification is relaxed. If feedback is given

to the system about the correctness of its answers, the posed question and the returned correct answer are used to feed a new pattern learning cycle.

In the evaluation we compare our approach with classical AE strategies. Results show that our approach contributes to the system's best achieved performance when it is conjugated with the remaining AE strategies. Moreover, our approach allows the system to learn with posed questions and is able to rectify past questions.

This paper is organized as follows: Section 2 refers to related work; Section 3 describes the approach; Section 4 presents the evaluation and Section 5 highlights the main conclusions and points to future work.

2 Related Work

The pattern-based approach to AE has received much attention in QA, specially after the TREC-10 conference, where the winning system (Soubbotin, 2001) relied on a list of hand-built surface patterns.

The focus has turned to automatic pattern learning, with many approaches taking advantage of the large amount of information on the Web. For example, Ravichandran and Hovy (2002) and Zhang and Lee (2002) learn surface patterns for a particular relation. Javelin (Shima and Mitamura, 2010) adds flexibility to the learned patterns by allowing, for instance, certain terms to be generalized into named entities. In all the previous cases, seeds are sets of two (manually chosen) entities. On the contrary, Ephyra system (Schlaefer et al., 2006) takes as input questions and answers, and learns patterns between the answer and two or more key-phrases. However,

these are extracted from the question by hand-made patterns. Previous approaches learn lexical patterns; others learn patterns also conveying syntactic information in the form of dependency relations (Shen et al., 2005; Bouma et al., 2011). Particularly, Shen *et al.* define a pattern as the smallest dependency tree that conveys the answer and one question term.

3 From Answered Questions to QA

3.1 JUST.ASK – An Open Domain QA System

JUST.ASK is the open-domain QA system that we will use in all our experiments. It follows the typical pipelined architecture of QA systems composed of three main stages: Question Processing, Passage Retrieval and AE.

In the Question Processing stage, the Berkeley Parser (Petrov and Klein, 2007), trained on the QuestionBank (Judge et al., 2006), analyzes each question. A machine learning-based classifier (SVM) fed with features derived from a rule-based classifier (Silva et al., 2011) classifies the questions according to Li and Roth's taxonomy (2002). Concerning the Passage Retrieval stage, the relevant passages are retrieved from the information sources (local or the Web). Finally, regarding the AE stage, JUST.ASK relies on a panoply of strategies and resources, namely: matching of regular expressions, machine learning Named Entity Recognition (NER) (by using the Stanford's Conditional Random Field-based recognizer (Finkel et al., 2005)), gazetteers, dictionaries constructed *on the fly* and the linguistic patterns previously learned. We will call NER to the combination of all the implemented AE strategies, except the pattern-based strategy.

3.2 Pattern Learning in a Nutshell

To learn patterns, we follow the approach described in (omitted, 2011). Briefly, lexico-syntactic patterns are learned from Question/Answer (Q/A) pairs (each composed of a question and its answer) and from sentences retrieved from the information sources that contain constituents of these pairs. Each pattern relates a question to its answer, through the constituents of a sentence that contains parts of the question and the answer. For example, the pattern

"NP_{answer} has VBN NP"[1] is learned from the sentence *Dante has written The Divine Comedy* and the Q/A pair (*Who wrote The Divine Comedy?*, *Dante*).

All the learned patterns and the source Q/A pairs are stored in a Knowledge Base (KB).

3.3 Pattern-based Answer Extraction

Figure 1 depicts the flow of information of the pattern-based AE strategy with feedback loop.
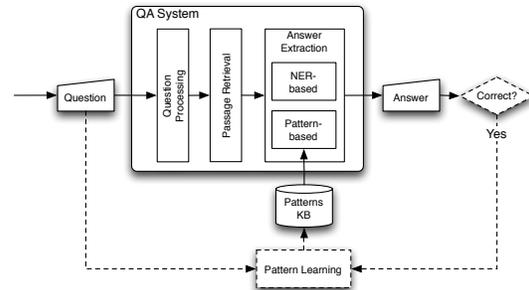

Figure 1: Learning from answered questions in JUST.ASK.

Given a new question, its syntactic analysis (performed in the Question Processing stage) allows to retrieve from the KB all the applicable patterns. Relevant passages are returned in the Passage Retrieval stage and, in the AE stage, there is a pattern/sentence unification to extract the candidate answers: every sentence in the passages is analyzed and its parse tree is explored in a top-down, left-to-right, depth-first search, unifying the obtained sentence tree components with the lexico-syntactic information in the patterns.

Since the pattern-learning algorithm receives a Q/A pair as input, if the system is able to verify the correctness of its answers, the process to learn new patterns is straightforward. Therefore, if positive feedback is given, JUST.ASK learns new patterns with previously answered questions.

3.4 Lexico-Syntactic Relaxation Strategies

Due to the variability of natural language, a pure pattern/sentence unification is rare. Therefore, the unification process relies in the following:

1. If there are no patterns in the KB that apply to the question – if the system has not yet

---

[1] Note that the pattern "NP_{answer} has VBN NP" indicates that the answer is in the first NP.

seen a question with the same syntactic structure and semantic category of the new question: JUST.ASK runs the NER strategies.

2. If there are patterns in the KB that apply to the question, but the unification does not succeed and no answer is captured, JUST.ASK uses two relaxation strategies: a) A lexical distance (Levenshtein, overlap and Jaccard) is permitted between the sentence and the question lexical components. This allows, for instance, to cope with orthographic errors. b) A syntactic match is consented between (syntactic) classes that relate with each other as they belong to the same (supper) class (*e.g.*, both "NP" (proper noun) and "NNS" (plural noun) are nouns). This allows the pattern "NP has VBN NP" to match a sentence analyzed as "NN has VBN NP".

4 Evaluation

4.1 Experimental Setup

The evaluation is made with a corpus of questions and their correct answers, built from the freely available data of the TREC QA tracks, years 2002 to 2007, where the answers were given by the competing systems and judged as correct by human assessors. From the original corpus, we collected the questions whose answers can be found in the New York Times (NYT) and discarded those with pronominal anaphora and ellipsis. The resulting corpus contains 1000 Q/A pairs. Lucene[2] was used to index and retrieve the paragraphs from the NYT from 1987 to 2007.

We assume that the selection of the final answer is flawless: if JUST.ASK is able to find a correct answer, it will retrieve it. To simulate the feedback loop, the answers of the corpus are used: if JUST.ASK finds an answer from the corpus, the learning process begins, as a (human) positive feedback would do.

Performance is measured using precision, recall and F-measure (Jurafsky and Martin, 2000). Results are calculated as a function of the number of questions previously evaluated (when evaluating the ith question P_i, R_i and F_i are determined).

In a first experiment, we test our approach in 4 scenarios, where we vary the strategies employed for AE: 1) *NER*: JUST.ASK runs with the NER strategies; no (pattern) learning process is involved; 2) *Patterns, with reference as fallback*: the pattern-based strategy is used. If no correct answer is found, one answer from the corpus is chosen and used (with the posed question) as input to a new pattern learning cycle. By doing so, we mimic the behavior of a tutor, which explicitly provides the correct answer to the system;[3] 3) *NER and Patterns*: combination of all implemented AE strategies; 4) *NER and Patterns, with reference as fallback*: combination of all implemented AE strategies, but if no answer is found, an answer from the corpus is used.[3]

In a second experiment, we allow JUST.ASK to revise wrong or unsolved past questions and try to solve them using recent knowledge. Thus, we stop the interactions at the $(i * n)^{th}$ question ($i = \{100, 250, 500\}$, $n \in \{1, ..., 10\}$ and $i * n < 1000$) and try to answer all previous wrong or unsolved questions, by using the new learnt patterns (except the ones learnt with the question in revision).

4.2 Results

Concerning the first experiment, results at the 1000th question are in Table 1.[4]

	Scenario			
	1	2	3	4
R_{1000}	45.1%	12.7%	46.4%	**47.5%**
P_{1000}	79.2%	43.5%	**79.9%**	77.2%
F_{1000}	57.5%	19.7%	57.8%	**58.8%**

Table 1: Results of the first experiment.

Table 2 reports the results of the second experiment at the 1000th question.

Considering the first experiment, best results (recall and F-measure) are achieved when the pattern based approach is conjugated with NER. Precision attains best results in scenario three.

Considering the second experiment, JUST.ASK's recall increases when i decreases: the more backward loops, the more correct answers are extracted. We should mention that, for instance, when

| | **No** | **i** | | |
	Reiteration	100	250	500
R_{1000}	12.7%	**17.7%**	16.8%	15.2%
P_{1000}	43.5%	**47.8%**	46.3%	45.4%
F_{1000}	19.7%	**25.8%**	24.7%	22.8%

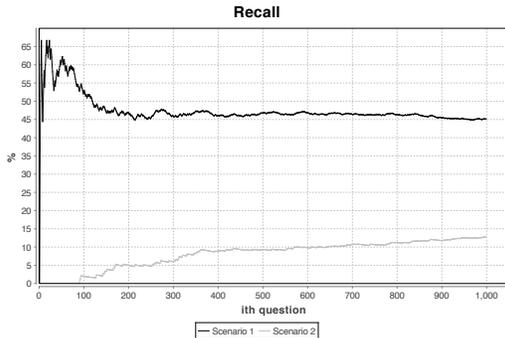Table 2: Results of the second experiment.

Figure 2: Evolution of recall with the posed questions

JUST.ASK stops at the 500th question, it correctly answers 25 questions that it was not able to solve before. Moreover, we verify that if all patterns learned with the 1000 questions are employed, JUST.ASK can solve 188 of the total 1000 questions (again the patterns learnt from a question are not used to answer it).

Figure 2 allows us to compare the recall evolution on scenarios one and two. It can be seen that, although leading to the worst results, there is a clear upward tendency of the recall in the second scenario, which is not verified in the scenario where NER is used. Moreover, from the same figure, we can see that, in the second scenario, the recall of JUST.ASK starts to increase around the 90th question and keeps on rising with the number of posed questions. This is because the system learns new patterns and correctly answers more questions.

We should add that we observed much overlap between the remaining scenarios (not shown on the figure). However, the recall becomes slightly higher with the number of answered questions on the third and fourth scenarios when compared with the first one, where JUST.ASK is not learning.

Finally, regarding the relaxation strategies, 105 of the 127 correct answers were extracted using an exact unification, as 22 questions were correctly an-

swered when using the relaxation strategies.

4.3 Analysis

Looking at the achieved results, the first fact that needs to be mentioned is that, although we evaluate our system with 1000 questions, only 791 were answerable since Lucene was able to retrieve passages containing the answers for 791 questions only.

Moreover, we find out that patterns aid JUST.ASK's performance by discovering answers to misclassified questions. For instance, the questions *Who did France beat for the World Cup?* expects a country as an answer. However, since it is classified as requiring the name of an individual, the correct answer *Brazil* is not extracted by the NER strategies. Patterns were successful in this situation and also to find answers to questions like *What is the "Playboy" logo?* and *How did Malcolm X die?*, for which JUST.ASK does not have a specific NER strategy.

To conclude, we believe that the diversity of the questions posed to the system and the small amount of information used to extract answers lead to the significantly poor results of the pattern-based strategy. However, results show increasing performance along the learning process, suggesting that this approach would benefit from more seeds and broader information sources. Due to the lack of available resources that make this experiment reproducible (either questions and answers, and information sources where the answers are stated), we could not empirically verify the upper bound of recall.

5 Conclusions and Future Work

In this paper, we proposed a fully-automatic pattern-based approach to AE, in which the system is itself the provider of the instances (question/answer) that feed the pattern learning algorithm. After new patterns are learned, they are immediately available to extract new answers. To the best of our knowledge, JUST.ASK is the first system able to learn from successfully answered questions as well as to revise and amend previous questions with newly acquired knowledge.

As future work, we want to learn to score patterns, rewarding the ones that extract correct answers and penalizing those that do not extract any correct can-

didate answer. Another planned task is to implement other relaxation strategies in the pattern unification.

References

Gosse Bouma, Ismail Fahmi, and Jori Mur. 2011. Relation extraction for open and closed domain question answering. In *Interactive Multi-modal Question-Answering*, Theory and Applications of Natural Language Processing, pages 171–197. Springer Berlin Heidelberg.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL '05: Proc. 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. ACL.

John Judge, Aoife Cahill, and Josef van Genabith. 2006. Questionbank: creating a corpus of parse-annotated questions. In *ACL-44: Proc. 21st Int. Conf. on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 497–504. ACL.

Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition.

X. Li and D. Roth. 2002. Learning question classifiers. In *Proc. 19th Int. Conf. Computational linguistics*, pages 1–7. ACL.

omitted. 2011. Omitted for blind review purposes. In *Omitted for blind review purposes*.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proc. Main Conference*, pages 404–411. ACL, April.

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *ACL '02: Proc. 40th Annual Meeting on Association for Computational Linguistics*, pages 41–47. ACL.

Nico Schlaefer, Petra Gieselmann, and Thomas Schaaf. 2006. A pattern learning approach to question answering within the ephyra framework. In *Proc. 9th Int. Conf. on TEXT, SPEECH and DIALOGUE*.

Dan Shen, Dietrich Klakow, and Geert-Jan Kruijff. 2005. Exploring syntactic relation patterns for question answering. In *Proc. 2nd International Joint Conf. on Natural Language Processing*. Springer.

Hideki Shima and Teruko Mitamura. 2010. Bootstrap pattern learning for open-domain clqa. In *Proc. NTCIR-8 Workshop*.

João Silva, Luísa Coheur, Ana Mendes, and Andreas Wichert. 2011. From symbolic to sub-symbolic information in question classification. *Artificial Intelligence Review*, 35:137–154.

M. M. Soubbotin. 2001. Patterns of potential answer expressions as clues to the right answers. In *Proc. Tenth Text REtrieval Conference (TREC*, pages 293–302.

Dell Zhang and Wee Sun Lee. 2002. Web based pattern mining and matching approach to question answering. In *TREC'02*.