Chinese Answer Extraction Based on POS Tree and Genetic Algorithm

Shuihua Li, Xiaoming Zhang, Zhoujun Li
State Key Laboratory of Software Development Environment Beihang University
Beijing 100191, China
brooklsh@buaa.edu.cn

Abstract

Answer extraction is the most important part of a chinese web-based question answering system. In order to enhance the robustness and adaptability of answer extraction to new domains and eliminate the influence of the incomplete and noisy search snippets, we propose two new answer exraction methods. We utilize text patterns to generate Part-of-Speech (POS) patterns. In addition, a method is proposed to construct a POS tree by using these POS patterns. The POS tree is useful to candidate answer extraction of web-based question answering. To retrieve a efficient POS tree, the similarities between questions are used to select the question-answer pairs whose questions are similar to the unanswered question. Then, the POS tree is improved based on these question-answer pairs. In order to rank these candidate answers, the weights of the leaf nodes of the POS tree are calculated using a heuristic method. Moreover, the Genetic Algorithm (GA) is used to train the weights. The experimental results of 10-fold cross-validation show that the weighted POS tree trained by GA can improve the accuracy of answer extraction.

1 Introduction

As mature information retrieval tools, search engines can satisfy most information needs of people. But, with the rapid growth of Internet data, search engines' weakness is being revealed gradually. Traditional search engines which use keywords as input and provide a long list of HyperText Markup Language (HTML) documents are convenient for machines to run. However, Question Answering (QA) systems which use natural language as input are convenient for human beings to communicate. Some QA systems directly employ well-built search engines for this task which are called web-based QA systems (Sun et al., 2014). Most web-based QA systems have three modules: 1) question analysis module to analyze the unanswered question and generate queries which are needed for search engines; 2) search snippets retrieval module to send queries which consist of keywords to search engines and then obtain search snippets from search engines; 3) answer extraction module to extract the final answer from these search snippets. Web-based QA systems compromise the merits of search engines and QA systems: 1) the existing mature search engines enable web-based QA systems to use the abundant data on the Internet; 2) web-based QA systems can communicate with people in natural language.

At present, there are less studies on chinese web-based QA than english web-based QA. Moreover, chinese web-based QA embodies many improvements on candidate answer extraction and ranking. We focus on answer extraction of chinese web-based QA system which can answer factoid questions. In order to enhance the robustness and adaptability of answer extraction to new domains and eliminate the influence of the incomplete and noisy search snippets, we propose two answer exraction methods based on POS tree.

The similarities between questions are used to select the question-answer pairs whose questions are similar to the unanswered question. These question-answer pairs are utilized to generate text patterns which can be transformed to POS patterns. Then, we propose a method to construct a POS tree using these POS patterns. The POS tree is of use to candidate answer extraction of web-based QA. To rank candidate answers, we use a heuristic method to calculate the weights of the

Proceedings of the 9th SIGHAN Workshop on Chinese Language Processing, pages 30–36,
Taipei, Taiwan, December 1, 2017.

POS tree's leaf nodes. Moreover, the Genetic Algorithm (GA) (Andrew, 1993) is used to train the weights. The results of the experiments show that the weighted POS tree trained by GA can improve the accuracy of answer extraction.

Our contributions in the paper are three-fold: 1) proposal of a new chinese answer extraction method based on POS tree; 2) proposal of a training method for POS tree by using GA; 3) empirical verification of the proposed methods.

The remainder of this paper is organized as follows: in section 2, we will discuss about related work. In section 3, a method to construct a POS tree and another method to train the POS tree with GA will be presented in detail. In section 4, the experimental results of 10-fold cross-validation will be shown. In section 5, this paper will be concluded.

2 Related Work

Answer extraction is the most difficult part of a web-based QA system. As such, it is also the focus of this paper.

Traditional web-based QA systems typically use search snippets directly (Brill et al., 2001; Sun et al., 2015). Although plain texts in the retrieved HTML documents can offer more information (Ravichandran and Hovy, 2002; Liu et al., 2014), the search snippets as high-quality summarizations generated by search engines can save web-based QA systems from having to crawl, parse and filter HTML documents. In spite of efficiency improved by search engines, they lead to another problem: some state-of-the-art answer extraction methods (Severyn and Moschitti, 2013; Yao et al., 2013; Liu et al., 2014) rely on syntactic information cound be seriously affected by these search snippets which consis of incomplete sentences.

The process of extracting a final answer from the search snippets has two steps: 1) extract candidate answers such as names, dates and places, and so on; 2) rank these candidate answers based on ranking method to find the best one as the final answer. There are many candidate answer extraction methods, such as: 1) some work use dictionaries which are edited manually or generated automatically to generate candidate answers. For example, the famous QA system Watson (Chu-Carroll and Fan, 2011) extracts titles from Wikipedia entries as candidate answers. This method provides a large candidate answer set which requests lots of effort for maintaining, updating and ranking. Besides, this method has low adaptability to new domains. 2) The most commonly adopted method is to use Named Entity Recognition (NER) tools to extract Named Entity (NE) that matching with question type (Xu et al., 2003). This method is always used together with question type classification algorithm. The performance of this method will be limited by the performance of classification algorithm and NER tools. 3) Another commonly used method is to etract candidate answers with text patterns which are edited manually or generated automatically (Zhang and Lee, 2002; Bhagat and Ravichandran, 2008; Khashabi et al., 2016). This method has high precision. However, these text patterns are too fine-grained to be adapted to new data.

There are also many ranking methods which can choose a best answer from a candidate answer set, such as: 1) a simple and commonly used method is to rank candidate answers by the similarities between candidate answers and the unanswered question in Vector Space Model (VSM). This method can be used with Latent Semantic Analysis and word2vec tool (Mikolov et al., 2013). 2) Another commonly used method is to compute the similarities by syntactic information. To improve performance of this method, tree edit distance (Severyn and Moschitti, 2013) and factor graph (Sun et al., 2013) can be used. 3) Some work rank candidate answers by a combination of features, e.g., lexical features, semantic features, statistical features and similarity features, and so on (Severyn and Moschitti, 2013; Khodadi and Abadeh, 2016). For comprehensive utilization of these features, some global optimization algorithms such as GA are needed (Figueroa and Neumann, 2008).

3 Method

We have implemented a chinese web-based QA system. In this section, we will discuss our answer extraction method of the system in detail below. The method contains three steps: 1) construct a POS tree using POS patterns generated by the question-answer pairs whose questions are similar to the unanswered question; 2) train the weights of the leaf nodes of the POS tree; 3) extract and rank candidate answers with the trained POS tree to find the best answer.

Item Name	Item Value
question Q	北大校长是谁?
keywords of Q	北大, 校长
answer A	林建华
search snippet S	...中组部宣布林建华担任北大校长，王恩哥不再担任北大校长...
target substring S^*	林建华担任北大校长
segmentation of S^*	林建华/nr 担任/v 北大/j 校长/n
POS pattern P	nr#a v j#k n#k#e

Table 1: An example of extraction of a POS pattern.

3.1 POS Tree

Extension of POS: Given a word w, define $t(w)$ as its extension of POS. In addition to POS, $t(w)$ may have some of three different marks: 1) mark #a means w is a part of the answer; 2) mark #k means w is a keyword of the question; 3) mark #e means w is the last word of a pattern.

POS Pattern: Given an answered question Q and its answer A, if there is a search snippet S contains A and some keywords of Q, then there is a shortest substring S^* of S also contains A and some keywords of Q. We name S^* as target substring. If segmentation of S^* is $(s_1, s_2, ..., s_n)$, then we get a POS pattern $P = (t(s_1), t(s_2), ..., t(s_n))$.

POS patterns that are abstracted from text patterns have better adaptability. An example of exraction of a POS pattern is shown in Table 1. The extract POS pattern algorithm is shown in Algorithm 1.

Algorithm 1 Extract POS Pattern

Input: question's keywords K, answer A and search snippet S
Output: POS pattern P

```
if S contains K and A then
    S* ⇐ target substring of S
    P ⇐ ()
    for each word w in S* do
        append t(w) to P
    end for
end if
```

POS Tree: Given a POS pattern set L, we can construct a POS tree T which cover every POS pattern of L. T consists of extension of POS but excudes the root node. Every path from the root node to a leaf node in T represents a POS pattern in L.

To construct a POS tree, we need some question-answer pairs whose questions are similar to the unanswered question, because we believe that the more similar a couple of questions are, the more likely they will both match a POS pattern. To find these question-answer pairs, we transform questions to vectors with word2vec tool, then classify the unanswered question with Support Vector Machine (SVM) (Suykens and Vandewalle, 1999) and compute its cosine similarties between questions of all question-answer pairs of its category. In addition, the POS tree can not be cached, but those POS patterns can. Those cached POS patterns can speed up the construction of another POS tree. An example of a POS pattern set is shown in Table 2. For this example, the POS tree we can construct is shown in Figure 1(a). The construct POS tree algorithm is shown in Algorithm 2.

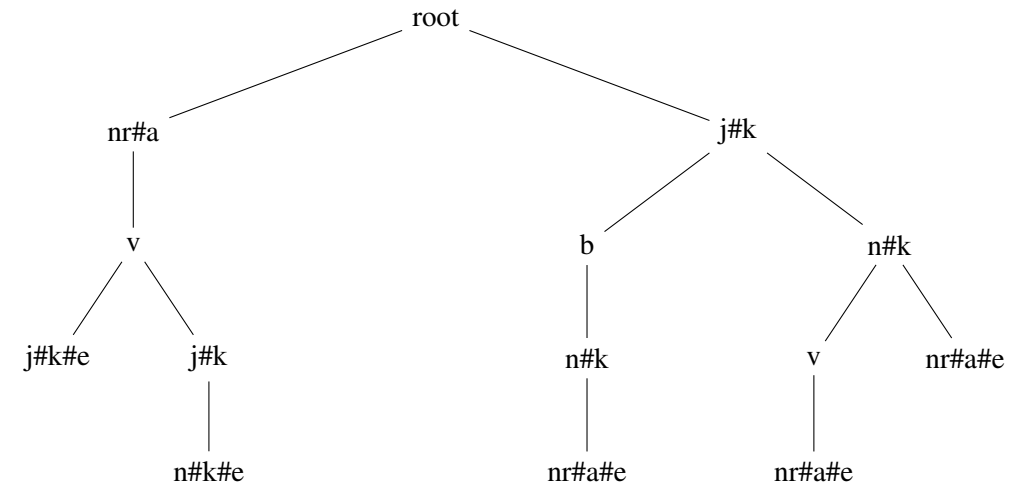
Target Substring	POS pattern
林建华担任北大校长	nr#a v j#k n#k#e
林建华挂帅北大	nr#a v j#k#e
北大新任校长林建华	j#k b n#k nr#a#e
北大校长是林建华	j#k n#k v nr#a#e
北大校长林建华	j#k n#k nr#a#e

Table 2: An example of a POS pattern set.

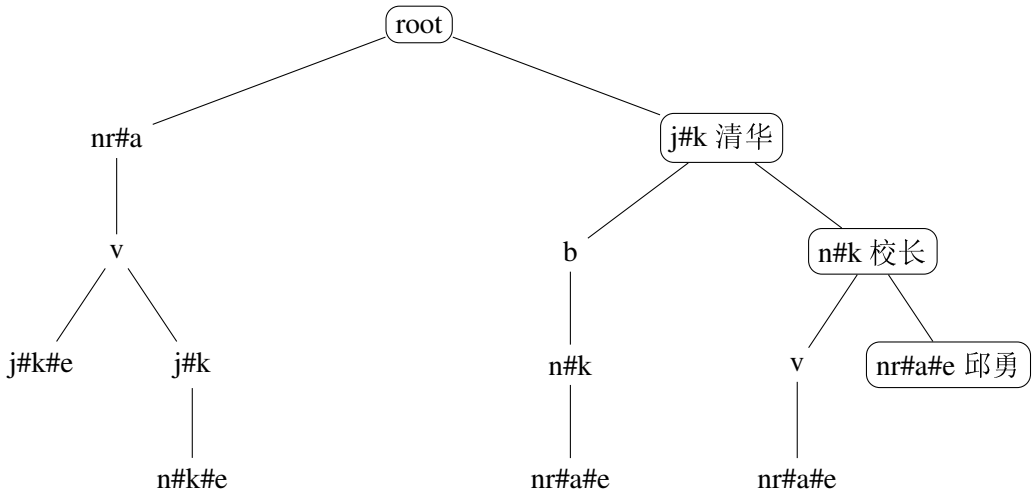
Algorithm 2 Construct POS Tree

Input: POS pattern set L
Output: POS tree T

```
add root node N_ROOT to T
for P ∈ L do
    N_NOW ⇐ N_ROOT
    for each t(w) in P do
        if N_NOW doesn't has a t(w) child then
            create a t(w) child for N_NOW
        end if
        N_NOW ⇐ the t(w) child of N_NOW
    end for
end for
```



(a) Before the answer extraction process.



(b) During the answer extraction process.

Figure 1: An example of a POS tree.

3.2 Candidate Answer Extraction Based on POS Tree

When a new question Q is submited, we can extract its keywords K and get a search snippet set X about it. For each search snippet S in X, the segmentation of S can be used to extract candidate answers with the POS tree T that we have constructed before.

For the example question Q in Table 3 and the POS tree T in Figure 1(a), we can extract candidate answer like Figure 1(b) and then we get a candidate answer "邱勇".

3.3 Train POS Tree and Rank Candidate Answers

In the previous subsection, we discussed how to extract candidate answers with a POS tree. However, people only need a best answer instead of many equally important candidate answers. To rank these candidate answers, the weights of the leaf nodes of the POS tree are calculated. The score of a candidate answer would be the sum over all the weighs of the leaf nodes which contribute to the generation process of this candidate answer. Then we rank these candidate answers by their score to choose the final answer.

Every leaf node of the POS tree corresponds to a POS pattern. So, there is a simple heuristic method to calculate weights of these leaf nodes: just set the weight of a leaf node to the number of POS patterns it corresponds to. These POS patterns are extracted from the question-answer pairs while we are constructing the POS tree. This method does work well. The experimental results of this method will be shown in the next section.

In addition, we use GA to train the weights of

Item Name	Item Value
question Q	清华校长是谁?
keywords of Q	清华, 校长
search snippet S	...2016年，清华校长邱勇毕业致辞...
segmentation of S	...2016/m 年/q ，/w 清华/j 校长/n 邱勇/nr 毕业/v 致辞/v ...

Table 3: A new question.

Algorithm 3 Train POS Tree

Input: POS tree T and similar question set V of the new question
Output: trained POS tree T

initialize population P_{NOW} which consists of random genes, every gene is a weight array for a leaf node of T.
$G_{BEST} \Leftarrow$ a random gene
while the number of iterations is less than the threshold **do**
  for $G \in P_{NOW}$ **do**
    set the weights of the leaf nodes of T to G
    set the fitness of G to MRR which computed using T and V
    if the fitness of G is higher than the fitness of G_{BEST} **then**
      $G_{BEST} \Leftarrow G$
    end if
  end for
  if the fitness of G_{BEST} is equals to the max fitness **then**
    break while
  end if
  $P_{NEXT} \Leftarrow \emptyset$
  while $|P_{NEXT}| < |P_{NOW}|$ **do**
    get two gene G_1 and G_2 by roulette selection from P_{NOW}
    cross or mutate G_1 and G_2 in a certain probability
    add G_1 and G_2 to P_{NEXT}
  end while
  $P_{NOW} \Leftarrow P_{NEXT}$
end while
set the weights of the leaf nodes of T to G_{BEST}

the leaf nodes. Every gene that used in GA is a weight array. Train data of GA is those question-answer pairs which are used to construct the POS tree. The fitness function of GA is Mean Reciprocal Rank (MRR) of the ranked candidate answers which are extracted from these question-answer pairs with the POS tree and a gene. The MRR is the average of the reciprocal ranks of the ranked candidate answers for n question-answer pairs:

$$MRR = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{rank_i} \quad (1)$$

where $rank_i$ refers to the rank position of the first relevant candidate answer for the i-th question-answer pair. The train pos tree algorithm is shown in Algorithm 3.

While the web-based QA system is constructing a POS tree, it can retrieval search snippets for the unanswered question at the same time. When the POS tree and these search snippets are both ready, the best answer can be extracted from these search snippets with the POS tree. Then, the system will return a best answer or top k ranked candidate answers with sentences based on the circumstances.

4 Experiments

Finally, in order to verify the effectiveness of our methods, we have built a question-answer dataset with 256 question-answer pairs artificially. There are five question types in this dataset: WHO,

Method	MRR of WHOs	MRR of WHENs	MRR of WHEREs	MRR of HOW MANYs	MRR of WHATs	MRR
NER	**0.6965**	0.4130	0.5610	0.5681	0.3102	0.5911
Simple POS tree	0.6943	0.6621	0.6319	0.4621	**0.5762**	0.6538
POS tree & GA	0.6051	**0.7422**	**0.7226**	**0.5758**	0.5458	**0.6615**

(a) On baidu data

Method	MRR of WHOs	MRR of WHENs	MRR of WHEREs	MRR of HOW MANYs	MRR of WHATs	MRR
NER	0.6756	0.3676	0.4762	0.4696	0.3444	0.5431
Simple POS tree	**0.6780**	0.6207	0.6144	0.4697	0.5213	0.6334
POS tree & GA	0.6053	**0.6579**	**0.6986**	**0.5182**	**0.5833**	**0.6409**

(b) On bing data

Table 4: The results of 10-fold cross-validation

Question/Method	WHO	WHEN	WHERE	NUM	WHAT
Question	洛神赋 是谁写的	平安夜是 什么时候	泰山在 哪个省	金庸写了 多少部小说	熊猫 吃什么
NER	1.曹植 2.甄姬 3.甄洛	1.2015年 2.2014年 3.12月24日	1.华山 2.山东 3.泰安	1.一 2.15 3.几	1.大熊猫 2.竹子 3.熊猫兔
Simple POS tree	1.曹植 2.顾恺之 3.张渊书	1.12月24日 2.12月25日 3.11月28日	1.山东 2.山东省 3.黄山	1.15 2.一 3.十四	1.竹子 2.粪便 3.杂食
POS tree & GA	1.曹植 2.顾恺之 3.甄宓	1.12月24日 2.12月25日 3.2016年12月24日	1.山东 2.山东省 3.泰安市	1.15 2.14 3.十四	1.竹子 2.又名 3.观赏鱼

Table 5: Some examples of experimental results.

WHEN, WHERE, HOW MANY, WHAT. For every question-answer pair, we have retrieved 100 search snippets from two popular search engines, baidu and bing.

In this paper, we experiment our two methods compared with the commonly used method, NER based method. The results of 10-fold cross-validation on baidu data is shown in Table 4(a) and on bing data is shown in Table 4(b). The heuristic method which is proposed in previous section is named "Simple POS tree", and the method with GA is named "POS tree & GA" in experimental results. From Table 4(a) and Table 4(b), we could see that our methods are better than the NER based method expect the WHO questions. We think the cause might be that the NER based method is good at name recognition but weak in recognition of other categories. The experimental results also show that GA can imporve the POS tree method. Our methods' performance on some pretty specific questions are shown in Table 5.

5 Conclusion

Web-based QA systems can extract a final answer from search snippets which are retrieved from search engines for an unanswered question. Answer extraction is the most important and difficult part of a chinese web-based QA system, because there are many incomplete and noisy sentences in these search snippets. In order to enhance the robustness and adaptability of answer extraction to new domains and eliminate the influence of the incomplete and noisy search snippets, we propose two new answer exraction methods. We utilize text patterns to generate POS patterns, then use POS patterns to construct a POS tree. The POS tree can be used to extract candidate answers from these search snippets. To rank these candidate answers, we propose a heuristic method and another method with GA. The results of 10-fold cross-validation show that the two methods work well and the weighted POS tree that trained by GA can improve the accuracy of answer extraction.

References

Alex M. Andrew. 1993. *Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, by john h. holland MIT press (bradford books), cambridge, mass., 1992, xiv+211 pp. (paperback £13.50, cloth £26.95). *Robotica*, 11(5):489.

Rahul Bhagat and Deepak Ravichandran. 2008. Large scale acquisition of paraphrases for learning surface patterns. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 674–682.

Eric Brill, Jimmy J. Lin, Michele Banko, Susan T. Dumais, and Andrew Y. Ng. 2001. Data-intensive question answering. In *Proceedings of The Tenth Text REtrieval Conference, TREC 2001, Gaithersburg, Maryland, USA, November 13-16, 2001*.

Jennifer Chu-Carroll and James Fan. 2011. Leveraging wikipedia characteristics for search and candidate generation in question answering. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*.

Alejandro Figueroa and Günter Neumann. 2008. Genetic algorithms for data-driven web question answering. *Evolutionary Computation*, 16(1):89–125.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Peter Clark, Oren Etzioni, and Dan Roth. 2016. Question answering via integer programming over semi-structured knowledge. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 1145–1152.

Iman Khodadi and Mohammad Saniee Abadeh. 2016. Genetic programming-based feature learning for question answering. *Inf. Process. Manage.*, 52(2):340–357.

Zengjian Liu, Xiao-Long Wang, Qingcai Chen, Yaoyun Zhang, and Yang Xiang. 2014. A chinese question answering system based on web search. In *2014 International Conference on Machine Learning and Cybernetics, Lanzhou, China, July 13-16, 2014*, pages 816–820.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

Deepak Ravichandran and Eduard H. Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pages 41–47.

Aliaksei Severyn and Alessandro Moschitti. 2013. Automatic feature engineering for answer selection and extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 458–467.

Hong Sun, Nan Duan, Yajuan Duan, and Ming Zhou. 2013. Answer extraction from passage graph for question answering. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 2169–2175.

Hong Sun, Furu Wei, and Ming Zhou. 2014. Answer extraction with multiple extraction engines for web-based question answering. In *Natural Language Processing and Chinese Computing - Third CCF Conference, NLPCC 2014, Shenzhen, China, December 5-9, 2014. Proceedings*, pages 321–332.

Huan Sun, Hao Ma, Wen-tau Yih, Chen-Tse Tsai, Jingjing Liu, and Ming-Wei Chang. 2015. Open domain question answering via semantic enrichment. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 1045–1055.

Johan A. K. Suykens and Joos Vandewalle. 1999. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300.

Jinxi Xu, Ana Licuanan, Jonathan May, Scott Miller, and Ralph M. Weischedel. 2003. Answer selection and confidence estimation. In *New Directions in Question Answering, Papers from 2003 AAAI Spring Symposium, Stanford University, Stanford, CA, USA*, pages 134–137.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. Answer extraction as sequence tagging with tree edit distance. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 858–867.

Dell Zhang and Wee Sun Lee. 2002. Web based pattern mining and matching approach to question answering. In *Proceedings of The Eleventh Text REtrieval Conference, TREC 2002, Gaithersburg, Maryland, USA, November 19-22, 2002*.