Domain Independent Keyword Identification for Question Answering

by

Prathyusha Jwalapuram, Radhika Mamidi

in

21st International Conference on Asian Language Processing
(IALP-2017)

Singapore

Report No: IIIT/TR/2017/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
December 2017
Domain Independent Keyword Identification for Question Answering

Prathyusha Jwalapuram  
Language Technology Research Center  
International Institute of Information Technology  
Hyderabad, India  
prathyusha.jwalapuram@research.iiit.ac.in

Radhika Mamidi  
Language Technology Research Center  
International Institute of Information Technology  
Hyderabad, India  
radhika.mamidi@iiit.ac.in

Abstract—In this paper, we look at domain independent keyword identification for natural language queries using statistical methods. We took queries supplemented by only their dependency tags (Stanford Parser) and part-of-speech tags (Stanford POS tagger) and labeled the keywords. We then delexicalised the training data, and used the Conditional Random Fields algorithm to learn these labels. We used the queries created by [1] in the course management domain for training, and tested our model on the queries of three domains: course management, library and the domain for training. Using the queries created by [1], we did this in a domain-independent manner, using only statistical methods. We took queries supplemented by only the tagset or for every new domain.

Keywords—keywords, constraints, queries, CRF++, domain independent;

I. INTRODUCTION

Keyword identification for question answering for a particular domain is usually done using keyword or pattern matching [2]. This domain specific approach requires the anticipation of a large number of patterns or keywords in order to cover all the possibilities and variations, and may still fail in the case of complex queries.

Using frames or patterns based on semantic grammar [3], [4] or a phrase and lexicon list [5] is also domain dependent and relies too much on pre-constructed lists, which suffer from the same issue of not being extensive enough to be practically useful.

General domain keywords might be identified for web scale data [6]–[8] which requires a large amount of training instances, and may not be specific for natural language queries.

The general strategy used in domain-independent question answering systems, typically on web-scale data such as community question answering forums, etc., classify questions based on question words and then query documents using the question and in the results returned, they detect entities to match with the answer [9]. These often failed in cases which had a person as answer without ‘who’ such as ‘name the person..’ or ‘which person’, and so on, showing that they had little semantic understanding of the query.

We approached the problem of keyword identification as a sequence labelling problem, which allows us to capture dependencies within the natural language query; we did this in a domain-independent manner, using only dependency relations and part-of-speech information, with a relatively small dataset. Section II describes the related work, Section III describes our data, Section IV explains our approach, Section V discusses our experiments and results and Section VI ends the paper with conclusions and future work.

II. RELATED WORK

[10] propose a rule-based system based on the framework of Computational Paninian Grammar [11] which identifies the semantic templates a query might belong to. Paninian grammar constructs are mapped to dependency relations, and verb frames with specified arguments for a particular domain are created. This limits the structures of the queries that can be processed; conversely, all the possible verb-argument structures must be identified for all possible verbs.

[12] model the keyword extraction for describing the meaning of a document in Chinese as string labeling. They use CRF [13] for the keyword extraction and show that it outperforms SVM and multi-linear regression. A large number of local and global features including a word window of +/- 2, length of the word, tf-idf, occurrence in title/abstract/full-text/reference, position of first appearance, and so on are used on 600 documents in the field of economics. They achieve a best F1 score of 0.5125.

[1] use the CRF algorithm [13] for concept identification in an NL query, which is an intermediary stage in a Natural Language Interface to Database (NLIDB) system. Here, concept refers to the tables, attributes and relations in a database schema; NL tokens in the query are mapped to one of these using a specific tagset, essentially reducing the concepts identification problem to a kind of Named Entity Recognition (NER) problem. The system is domain-specific to the course management domain, and requires annotation and training when there are changes made to the tagset or for every new domain.

[14] describe a rule-based system that is able to identify keywords in a domain independent manner. They use rules that select certain dependency tags over others as probable keywords and constraints. Using the keywords and constraints thus identified, they build a query equation that can then be converted to any form as appropriate for an ontology or an SQL query. We use a statistical approach for the keyword identification and compare the results.
III. DATA

We collected data from three different domains for training and testing purposes.

1) Course Management Domain: The dataset created by [1] consists of 1000 queries for training and 558 for testing. We used the dataset in the same way to make comparison possible.

2) Library Domain: We collected around 128 queries in the library domain through a survey. We used these queries for testing.

3) GEOQUERIES250: We use the GEOQUERIES250 dataset [4] for testing to facilitate comparison as it is a commonly used and well-known query dataset.

IV. APPROACH

Every word in the query is supplemented with its part-of-speech tag and dependency tag obtained from the Stanford POS tagger and the Stanford Parser respectively. The training data also includes the part-of-speech tag and the dependency tag of the parent word (in terms of dependency relation).

We use Conditional Random Fields to learn the labels since it provides a method to segment and learn sequence dependent labels, and has been shown to have advantages over HMMs and MEMMs [13].

A. Training Data

We use only dependency relations (for their syntacto-semantic information) and part-of-speech tags as part of external/meta information.

The data consists of current dependency tag $D_t$, current POS tag $P_t$, parent-POS tag $PP_t$ (POS tag of parent word through dependency relation) and a label indicating whether the current tag is to be selected (as a keyword) or discarded. The system will then use the corresponding features for processing, but the model itself is blind to the lexicon. The model can therefore predict labels for any set of dependency tags and POS tags belonging to any reference, the relationships between the keywords can be used to obtain important information to answer the query. For instance, from the example in Table I, the selected keywords and their dependency relations can be further structured like so:

```plaintext
posted(assignments, for_NLP)
```

This gives us some amount of semantic information that can be used to answer the query; it can be converted to SQL or any other form as required by the knowledge base for question answering.

B. Question Answering

We conduct some experiments in order to determine the best template (features) for the statistical model to learn from. If the current dependency tag is $D_t$, then the previous tag (dependency tag of previous word in the query) is $D_{t-1}$, the next tag (dependency tag of the next word in the query) is $D_{t+1}$, and so on. These features are in addition to the $D_t$, $P_t$ (current POS tag) and the $PP_t$ (current Parent-POS tag) that are used for every instance. A template having $x/y$ indicates a combined feature of $x$ and $y$.

We experiment with different templates in CRF++ to find the most optimised template for learning. The accuracies for different templates for the different domains are given in Table II. The precision, recall and F1 scores for each template and domain are given in Table III.

As highlighted in Table 2, template number 5 that uses combinations of two previous dependency and POS tags and one next dependency and POS tag (window of -2 to +1 for combined features) along with the current dependency, POS and parent POS tag performs best for the course management and library domains. Template number 4 that uses combined features of one previous dependency and POS tag (window of -1 to +1 for combined features) in addition to the previous, current and next dependency and POS and current parent POS tag (window of -1 and +1 for unigram features) performs the best for the GEOQUERIES250 dataset.

We can see that, in general, there is a correlation between a keyword, its dependency tag, its POS, the

![Table I: Example of Training Data](image)

1) Labeling: Each word in the query is manually labeled as SELECT or DISCARD based on whether it is a keyword important for answering the query or not.

2) Delexicalisation: In order to make our approach truly domain independent, the training data is delexicalised, i.e., stripped of all actual words. The input training data (before it is processed according to templates) therefore only consists of dependency tag and POS tag of current word and the dependency tag and POS tag of the parent word, plus the label, as in Table I (excluding the first column). The test data is similarly labeled. The algorithm actually learns the correlation between whether a certain dependency tag and POS tag, along with a few other features, are likely to be a keyword or not; the word itself is irrelevant.
dependency and POS of the previous word, the POS of the parent word in terms of dependency relations, and the dependency and POS of the next word.

VI. COMPARISON WITH OTHER SYSTEMS

In Table IV, we compare our best accuracies against the accuracies obtained by the rule-based system in [14]. We reach high accuracies of 90.65% on the course management domain dataset, 83.19% on the library domain dataset and 97.13% on the GEOQUERIES250 dataset.

Also, from Table III we see that we comfortably exceed the F1 score of 0.5125 for keyword extraction using CRF set by [12].

VII. ERROR ANALYSIS

We see that the model performs fairly well on the test set of the same domain it is trained on (course management), but dips for the library domain. The dataset of the library domain is small compared to the others, which could be a contributing factor. The accuracies for the GEOQUERIES250 dataset are quite high, which may be due to the simple structure and non-ambiguousness of the queries. The course management and library domains have a fair number of complex queries which have relative clauses or multiple verbs; such queries are likely to have a higher incidence of parsing errors, which may contribute to an error in keyword identification.

A dependency relation-wise analysis shows that the highest number of errors occurred in the labeling of the root (main verb), nssubj (nominal subject of verb) and the dobj (direct object of verb) relations. A fair number of the root errors occurred due to parsing errors leading to a wrongly tagged root. Other errors occurred because of ambiguity in quite a few queries where the main verb is a keyword (teach, register) and where the main verb is not a keyword (list, give).

Similarly, queries which have multiple nssubj and dobj or both contribute to the errors; these dependency tags are also most common for keywords. Also, in questions with a "What..." construction, what often gets tagged as the nssubj or dobj, also causing errors when it is wrongly labeled as a keyword.

Other miscellaneous errors involved some infrequent dependency relations such as xsubj, rcmod, etc.

VIII. CONCLUSIONS AND FUTURE WORK

We see that our system performs with a high accuracy in identifying the relevant keywords in all three different domains of course management, library and the GEOQUERIES250 dataset, using a training dataset of only 1000 queries. This makes our approach efficient, easy to implement and truly domain independent.

Because the tagset of the dependency relations and the part-of-speech tags are more or less universally agreed upon and are the only external information used in our system, our approach does not require hand-crafted features that is different for each domain and can be universally used for any domain.

There is no need for re-annotation and re-training for every new domain that is needed; an existing robust model trained on a fair sized dataset can perform very well on all domains. A one-off training and and its resultant model are therefore all that is required for keyword identification in any domain.

Since dependency tags are also arguably language independent, and will produce the same tags (adjusted for the tagset) for a similar sentence in any other language, our
### Table III

<table>
<thead>
<tr>
<th>Domain</th>
<th>CRF</th>
<th>RBS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course Management</td>
<td>90.65%</td>
<td>61.1%</td>
</tr>
<tr>
<td>Library</td>
<td>83.19%</td>
<td>72.72%</td>
</tr>
<tr>
<td>GEOQUERIES250</td>
<td>97.13%</td>
<td>-</td>
</tr>
</tbody>
</table>

\textbf{Table IV}

\textbf{Accuracy of our CRF Model Compared to the Rule-Based System}

The approach can also be a language independent solution in addition to being domain independent.

In order to make the approach both language and domain independent, universal dependency tags [15] that are consistent across several languages can be used.

The approach needs to be tested across more domains with more varied patterns of queries, especially complex queries with relative clauses/multiple verbs. Whether the approach can be extended to queries with multiple sentences/descriptive questions should also be explored. The system may also be able to produce labels with more granularity, e.g. main keyword, additional constraint, etc.

\textbf{References}


