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 GeoQueries250 dataset and report fairly high accuracies of 90.65%, 83.19% and 97.13% respectively, making our model a truly domain independent and highly accurate keyword identifier.**

***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 sentence, making it domain independent.

See Table I for an example training instance for the query "What are the assignments posted for NLP?", taken from the course management domain. The training data does not include the first column, i.e., the words in brackets are not part of training.

Word	D_t	P_t	PP_t	**Label**
(What)	dobj	WP	VBN	DISCARD
(are)	auxpass	VBP	VBN	DISCARD
(the)	det	DT	NNS	DISCARD
(assignments)	nsubjpass	NNS	VBN	SELECT
(posted)	root	VBN	root	SELECT
(NLP)	prep_for	NNP	VBN	SELECT

Table I
Example of Training Data

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 1 (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.

B. Question Answering

The labeled/identified keywords can be formed into a query equation [14]: using the dependency tags as a 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:

$$posted(assignments, for\_NLP)$$

. 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.

V. Experiments

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 combined features 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

Template No.	Unigram Features	Combined Features	Course Domain	Library Domain	GQ250 dataset
1	D_t P_t PP_T	- -	88.51	79.14	96.74
2	D_t P_t PP_T	$D_{t-1}/D_t, D_t/D_{t+1}$ $P_{t-1}/P_t, P_t/P_{t+1}$	90.58	78.08	96.87
3	D_t, D_{t+1} P_t, P_{t+1} PP_T	$D_{t-1}/D_t, D_t/D_{t+1}$ $P_{t-1}/P_t, P_t/P_{t+1}$	90.28	78.51	96.67
4	D_{t-1}, D_t, D_{t+1} P_{t-1}, P_t, P_{t+1} PP_T	$D_{t-1}/D_t, D_t/D_{t+1}$ $P_{t-1}/P_t, P_t/P_{t+1}$	90.36	76.17	**97.13**
5	D_t P_t PP_T	$D_{t-2}/D_t, D_{t-1}/D_t, D_t/D_{t+1}$ $P_{t-2}/P_t, P_{t-1}/P_t, P_t/P_{t+1}$	**90.65**	**83.19**	93.15
6	D_t P_t -	$D_{t-2}/D_t, D_{t-1}/D_t, D_t/D_{t+1}$ $P_{t-1}/P_t, P_t/P_{t+1}$	90.06	78.29	95.5
7	D_t P_t -	$D_{t-2}/D_t, D_{t-1}/D_t, D_t/D_{t+1}$ $P_{t-2}/P_t, P_{t-1}/P_t, P_t/P_{t+1}$	90.65	78.08	92.43
8	D_{t-2}, D_{t-1}, D_t, D_{t+1} P_{t-2}, P_{t-1}, P_t, P_{t+1} PP_T	$D_{t-1}/D_t, D_t/D_{t+1}$ $P_{t-1}/P_t, P_t/P_{t+1}$	90.21	75.95	96.28

Table II
FEATURE TEMPLATES AND ACCURACIES

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 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), *nsubj* (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 ambiguousness 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 *nsubj* 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 *nsubj* or *dobj*, also causing errors when it is wrongly labeled as a keyword.

Other miscellanous 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

Template No:		1	2	3	4	5	6	7	8
Course									
	Precision	0.9107	0.9223	0.9184	0.9223	0.9262	0.9262	0.9275	0.9223
	Recall	0.8922	0.9152	0.9137	0.9117	0.9132	0.904	0.9122	0.9094
	F1	0.9014	0.9188	0.9161	0.917	0.9197	0.915	0.9198	0.9158
Library									
	Precision	0.808	0.8348	0.8437	0.8214	0.9013	0.7964	0.8035	0.8482
	Recall	0.7702	0.742	0.744	0.7215	0.8204	0.7639	0.7563	0.7089
	F1	0.7886	0.7857	0.7907	0.7682	0.8589	0.7789	0.7792	0.7723
GEOQUERIES250									
	Precision	0.9643	0.9667	0.9619	0.9726	0.8906	0.9488	0.8882	0.9524
	Recall	0.9771	0.9771	0.9782	0.9761	0.9752	0.9696	0.9726	0.9804
	F1	0.9706	0.9719	0.97	0.9743	0.931	0.9591	0.9285	0.9662

Table III
PRECISION, RECALL AND F1 SCORES FOR EACH TEMPLATE AND DOMAIN

Domain	CRF	RBS
Course Management	90.65%	61.1%
Library	83.19%	72.72%
GEOQUERIES250	97.13%	-

Table IV
ACCURACY OF OUR CRF MODEL COMPARED TO THE RULE-BASED SYSTEM

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.

REFERENCES

[1] S. Srirampur et al, "Concepts identification of an NL query in NLIDB systems" in International Conference on Asian Language Processing, 2014, pp. 230-233. IEEE.

[2] T. Johnson, Natural language computing: the commercial applications, The Knowledge Engineering Review, vol. 1, no. 03, pp. 1123, 1984.

[3] B. A. Goodman and D. L. Waltz , "Writing A Natural Languge Database System", Proceedings of the 5th International Joint Conference on Artificial Intelligence, 1977, Volume 1 I, pp. 144-150.

[4] A. Popescu et al, Towards a theory of natural language interfaces to databases, in Proceedings of the 8th international conference on Intelligent user interfaces. ACM, pp.149-157, 2003.

[5] M. Minock, "A Phrasal Approach to Natural Language Interfaces over Databases", Proceedings of the International Conference on Applications of Natural Language to Information Systems (NLDB), pages 333-336, June 2005.

[6] O. Etzioni et al, "Methods for domain-independent information extraction from the web: An experimental comparison", in Association for Advancement of Artificial Intelligence, 2004, pp. 391-398.

[7] W. Gatterbauer et al, "Towards domain-independent information extraction from web tables", in Proceedings of the 16th international conference on World Wide Web, 2007, pp. 71-80, ACM.

[8] K. Eichler et al, "Unsupervised and domain-independent extraction of technical terms from scientific articles in digital libraries" in Lernen, Wissen & Adaptivit Workshop Proceedings, 2009, pp. WIR-13.

[9] E.M. Voorhees, "The TREC-8 Question Answering Track Report", in Text Retrieval Conference, 1999, Vol. 99, pp. 77-82.

[10] A. Gupta et al, "A Novel Approach Towards Building a Portable NLIDB System Using the Computational Paninian Grammar Framework", International Conference on Asian Language Processing, 2012.

[11] A. Bharati et al, "Parsing Paninian Grammar with nesting constraints", Proceedings of 3rd NLP Pacific Rim Symposium, Seoul, S. Korea, 1995.

[12] C. Zhang et al, "Automatic keyword extraction from documents using conditional random fields", Journal of Computational Information Systems, 2008, 4(3), pp.1169-1180.

[13] J. Lafferty et al, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data", 2001.

[14] P. Jwalapuram and R. Mamidi, "Keyword and Constraint Identification for Question Answering", in 15th International Conference of the Pacific Association for Computational Linguistics, Yangon, Myanmar, 2017, in print.

[15] J. Nivre et al, "Universal Dependencies v1: A Multilingual Treebank Collection", in International Conference on Language Resoures and Evaluation, 2016.