Question Answering System for the Travel Domain

Hasangi Kahaduwa, Dilshan Pathirana, Pathum Liyana Arachchi, Vishma Dias, Surangika Ranathunga
Dept. of Computer Science and Engineering
University of Moratuwa
Katubedda 10400, Sri Lanka
{hasangik.12, pathirana.12, pathum.12, vishma.12, surangika}@cse.mrt.ac.lk

Upali Kohomban
CodeGen International (PVT) Ltd,
Colombo 10, Sri Lanka
upali@codegen.co.uk

***Abstract*— A Question Answering (QA) system backed by a comprehensive and up-to-date knowledge base would be appropriate for travellers to satisfy their information needs. In this paper, a complete QA system is presented. It has two main phases: question identification (Expected Answer Type (EAT) identification) and searching the knowledge base (KB) to find the answer to the classified question. In QA systems, identification of the EAT of a question imposes some constraints when determining the possible answer. This paper presents the first study on semantic classification of questions into EATs in the travel domain. A new two-level taxonomy for the travel domain is introduced, along with a dataset annotated with the same. A machine learning approach is used for question identification, which gives very promising results even with the use of syntactic and semantic features. A rule-based approach is used for searching the KB to find the answer. An ontology serves as the KB of the QA system which is traversed using a Simple Protocol and RDF Query Language (SPARQL) query generated through the rule-based approach.**

Keywords*— *Question Answering; Question Classification; Expected Answer Type; Taxonomy; Question Base; Travel Domain

I. Introduction

Travellers tend to plan their trips ahead to overcome any regrets and get the maximum out of their travels. During planning, it is essential for them to get answers for their questions and gather information about their trips. The questions they have may vary from very general questions such as “What are the travel apps available in Sri Lanka?” to very specific questions such as, “What is the best hotel in Colombo to stay during Christmas?”. There are several sources such as forums, websites, and travel agencies that travellers could use to acquire necessary information and answers to their questions. However, due to the unavailability of a central source of information, information and answers they receive might be contradictory. Thus a QA system backed by a comprehensive and up-to-date knowledge base would be appropriate for travellers to satisfy their information needs.

There has been past literature on several travel domain related QA systems, however these systems only represent a specific area of the travel domain. For example, TRAINS [1] is a QA system for railroad freights, Mercury Flight Reservation QA system [2] is only for flight reservations.

According to a study by Yang et al. [3], most of the QA systems are focused on open domains. Some examples are AskMSR [4] and EBay [5]. However, QA systems focusing on open domains are inapplicable for the travel domain, because the KBs they use, do not have much travel-specific information.

In this paper, we present the first study on a QA system for the general travel domain. There are mainly two phases of the system as question identification and searching the KB to find the answer. The question identification phase uses a machine-learning-based question classification approach that classifies the question into one of the pre-defined categories EATs. In this phase, a travel domain related two-level taxonomy and a question base of 5000 travel related questions were created. The accuracy of the classifier gained 0.87 for the first level and 0.76 for the second level of taxonomy. In the second phase, with the EATs generated from the question identification phase and the user question, a SPARQL query is generated to retrieve the answer by traversing the ontology that serves as the KB. A rule-based approach is used for ontology traversing. This showed an accuracy of 0.67.

This paper is structured as follows. Section 2 discusses previous work related to QA system approaches for each of the two phases. Section 3 describes the methodology including question identification and searching the KB to find answers. Section 4 presents the evaluation and finally section 5 concludes the paper with a look into future work.

II. Related work

Past literature is discussed under the two main phases of the QA system: question identification and searching the KB.

A. Question Identification

In QA systems, identification of the EAT of a question imposes some constraints when determining the possible answer, thus increasing the probability of finding the answer to a particular question. Hence in question identification, the EAT for the user question is determined. Question classification is one of the most established approaches that find the EATs using machine learning algorithms among several other approaches such as logical form-based [22], and rule-based [23]. Algorithms such as Support Vector Machines (SVM), Random Forest, Naïve-Bayes and Decision Trees have shown high precision and recall values in question classification for general [7],[12], as well as domain-specific [7] question classification.

Question classification approach requires the presence of four aspects:

978-1-5090-6491-5/17/$31.00 ©2017 IEEE

1. A taxonomy to categorise questions
2. A question base (corpus)
3. A classification algorithm
4. A feature set

1) Taxonomy

There are mainly two types of taxonomy: flat taxonomy and hierarchical taxonomy. Li and Roth [6] built the first hierarchical taxonomy for question classification in open domain, which has been used for question classification systems colossally. As for domain-specific taxonomies, Chernov et al. [7] created a hierarchical taxonomy for the interactive quiz domain.

2) Question Base

The UIUC dataset had been used for several research such as Li and Roth [6], and Zhang and Lee [8] in question classification to train the classifier. This dataset consists of 5000 general questions, and a test set of 500 questions that was manually annotated under the taxonomy created by Li and Roth [6]. Due to its extensive usage in implementing question classifiers, this could be considered as a benchmark to validate the performance of classifiers.

Question base for the biographical facts about famous people [7] is an example for a domain-specific question base.

3) Classification Algorithm

Three classification techniques have been commonly used:

- SNoW architecture [6]
- Support vector machines (SVM) [9], [8], [10]
- Deep learning [11]

Among these techniques, the SVM technique had shown better accuracies for a limited dataset and gives the researchers more influence on the classifier since the features used could be chosen according to the domain.

4) Feature Set

Features used for question classification can be broadly categorized into syntactic and semantic features [12]. According to past literature [6], [8], many syntactic features such as part of speech (POS) tags, lexical words, head chunks, surface text of questions (both bag of words and bag of n-grams), and named entities (NEs) have been used for open domain question classifiers. But it has become the state of art to use a mixture of both syntactic and semantic features and it had shown better results for both open domain [12] and specific domains [7].

B. Searching the KB

A commonly used mechanism to implement a KB is by means of an ontology. There are several possible ways of implementing an ontology such as first-order logic approach and UML-based approach that is based on use cases [21]. Resource Description Framework (RDF) is a language used to implement ontologies, and SPARQL is a query language to query RDF data in order to traverse ontology.

According to Sneiders [13], there are several approaches of generating a triple or a SPARQL query from a user question:

- Natural Language Processing (NLP) approach
- Information retrieval (IR) approach
- NLP + IR approach
- Rule-based approach

The NLP approach converts the user question into a formal presentation of meaning such as first order logic, semantic networks, conceptual dependency diagrams or frame-based representations. This is considered as one of the most reliable techniques, yet it is very computationally intensive and expensive [13], [14]. The IR approach retrieves documents or passages from documents. This had shown good results for web-based KB systems such as, SMART [15] and FAQ Finder [16]. Though NLP + IR approach is one of the trending techniques, some research [17] was not able to obtain satisfying results. The rule-based approach is a well-established technique that had shown better accuracies than NLP + IR approach and NLP approach with less computational intensiveness [18], [19]. This approach uses a manually created rule-base to match the question with a question template and a query template.

III. Methodology

The system overview of the proposed QA system is shown in Fig. 1. Question identification phase generates an EAT respective to the user question, and searching the KB phase generates an answer for the user question using the user question and the EAT generated from the question identification phase.

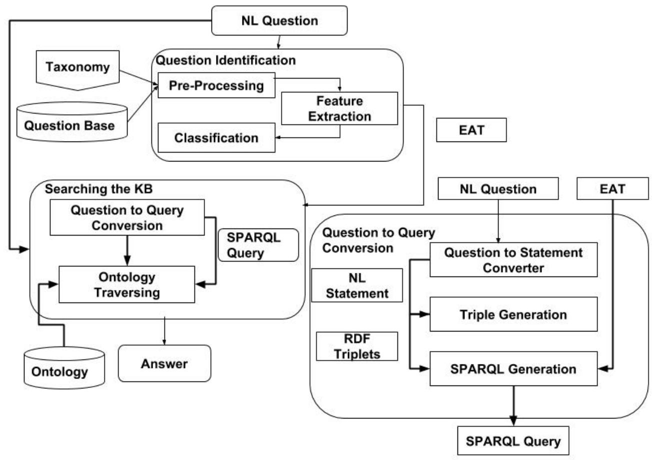

Fig. 1. QA System Overview

A. Question Identification

Since no related research could be found for question classification in the travel domain, all the four aspects of question classification had to be implemented.

1) Taxonomy

We created a taxonomy with 7 coarse classes and 63 fine classes that corresponds to an EAT specific to travel domain as shown in Table I. When creating this taxonomy, we used the basic categories (hotels, vacation rentals, flights, things to do and restaurants) available in TripAdvisor [24]. To overcome several ambiguities, we had to introduce new coarse classes than the above mentioned basic categories. Due to a wide range of information available in the travel domain, a flat taxonomy increased the number of coarse classes leading to many ambiguities. Hence, we adapted a hierarchical taxonomy. The fine level was developed considering questions and information related to travelling.

TABLE I. TWO LEVEL TAXONOMY FOR THE TRAVEL DOMAIN

Travel Guide	
Average costs & Expenses	Photographers, Videographers, Salons & Hairdressers, Wedding Planner
Banks & Money	Public Restrooms, Reading Areas
Cigarettes & Alcohol	Rules, Laws, Regulations & Taxes
Health & Safety	Telephones, GPS & Internet
Holidays & Open days	Tour Operators & Companies
Attire, Information Centers, Tipping, Language & Other information about the country	Transit, Airports, Duty Free
Laundry service	Travel related apps, WebSite & Books
Luggage storage & Lockers	Visa, Crossing Borders, Documents & Travel Insurance
Neighbourhood	Other
Transport	
Buses, Coaches & Trams	Flights, Helicopters & Airlines
Cabs, Taxis, Ubers, Songthaew & Tuk Tuks	Gas Stations
Car, Van, Scooter and Bike Rental	Routes, Distances, Time, Road Condition & Traffic
Cruises, Ferries, Boats	Trains, Subways, Metro, Shuttles & Cable cars
Drivers	Travel cards
Driving license & Self driving	Other
Food	
Authentic Food	Food Festivals
Breweries, Wineries & Liquor	Bakeries, Cafes, Coffee Shops & Restaurants
Catering & Delivery services	Bars & Pubs
Food Cost	Food Markets & Street Food
Food categories	Other
Accommodation	
Bungalows, Cottages, Farm houses, Campsites, Condos and Villas	Hotels & Motels
Caravans & RV Parks	Resorts
Beach Clubs & Guest Houses	Other
Apartments, Hostels & Private residence	
Things To Do	
Photography, Shopping & Tattoo	Streetwalks, Bus rides, Boat trips, Picnics, Side trips & Beach Walks
Sightseeing	Yoga, Meditation center & Gyms
Spa, Sun, Pools	Other
Sports & Adventurous activities	
Entertainment	
Festivals, Events & Parties	Shows, Parades, Operas, Theaters & Concerts
Live music, Karaoke, Jazz clubs, Blues, Comedy clubs	Sport events, Exhibitions
Nightlife & Dancing Clubs	Other
Weather	
Dusty, Rain, Chilly	Temperature
Snow	Other

2) Question Base

We created a question base of 5000 real world questions, which were collected from TripAdvisor [24] and Lonely Planet [25] travel forums. Manual reading of the forums and forming simpler questions out of complex questions or statements were essential during the creation of the question base. Most of the complex questions were restructured by splitting a question into multiple questions. This question base was then manually annotated for both coarse and fine class levels using the taxonomy given in Table I.

3) Classifiers

SVM was used as the classifier with a linear kernel function (linearSVM). During the implementation, it was essential to represent the non-numeric features as a bitmap since SVM only accepts numerical features.

We also experimented with Naïve-Bayes, decision trees, and random forest classifiers. However SVM outperformed all of these other techniques.

4) Features

All the features mentioned in past literature for both the open domain and specific domains were experimented with the SVM classifier. Moreover, we introduced WordNet semantic relatedness between an EAT and the syntactic head word and DBpedia category of an identified named entity as new features. Here, the latter is a domain dependent feature. Following shows the list of features we experimented with,

- bag of words - surface text words (ST) and POS-tags (POS)
- N-grams - unigrams (UG) and bigrams (BG)
- syntactic head word (Syntactic HW)
- Named Entities (NE)
- word shape (WS)

- Head Word (HW)
- Synonyms of Head Word (Synonyms of HW)
- WordNet Semantic relatedness
- DBpedia Categories (DBp)
- WordNet Synonyms

B. Searching the KB

In this phase, an answer is generated for the user question by searching in a KB. In this system, a proprietary ontology related to travel domain is used as the KB. This ontology is still under development, and currently it contains information related to hotels and things to do in Sri Lanka. This ontology was developed using the taxonomy created for the travel domain.

To traverse this, a SPARQL query generation is essential. Hence, it was necessary to derive a SPARQL query out of the user question. For this purpose, a rule-based approach is used. During this approach we first create RDF triples using a set of rules. A SPARQL query pattern is matched by the RDF triple along with the EAT generated from the first phase. To generate RDF triples, we used a different set of rules where the user question is matched to a RDF triple. However, RDF triple generation techniques we used (see Triple Generation) work with statements rather than questions. Hence it was required to convert the user question into a statement. This statement was then used to generate RDF triples.

1) Question to Query Conversion

This sub-phase takes the user question and the EAT generated from the question identification phase to create a SPARQL query. The process of this sub-phase is shown in Fig. 1.

a) Question to Statement Converter

This phase uses a rule-based approach for converting the natural language question to a statement. The QuestiontoStatementTranslator in the Stanford CoreNLP was used for this.

The statement creation process first POS tags and tokenizes the question and then matches it into a question template to categorize the question. Using manually written rules for each of the question templates, a statement is created. After identifying the matching question template, the question was re-structured into a statement using the tokens retrieved from POS-tagging.

For an example, consider the question, “Will there be any showers tonight in London?”. This question first goes through POS-tagging and results in the following expression “will/MD there/RB be/VB any/DT showers/NNS tonight/NN in/IN London/NNP ?/.?”. This expression is then used to match with the template given in the rule-base. The template that matches with this expression would be, “(?$modal [{tag:MD }]) (?$there [{lemma:there; tag:/EX|RB/}]) (?$be [{lemma:be}]) (?$statement_body []+?) (?$punct [word:/[?\\.!]/])”, which categorizes the question under the “Can/ Should/ Would/ Could/ Shall/ May/ Might/ Will be there” question template. After identifying the question template, the question was reconstructed using the manually written rules as “there will be any showers tonight in London”.

The dataset had a variety of question types that were not compatible with the QuestiontoStatementTranslator in Stanford CoreNLP. The 12 rules available there did not help to cover the whole dataset. For an example consider the question, “Is self-driving a good option in Manhattan?”, this question cannot be restructured to a statement using any of those 12 rules. Hence it was essential to recognize new patterns and write new rules to transform the recognized question patterns into statements. We introduced 24 new rules to this rule base. The new rules shown in Table II were written after a research on the question patterns in English language and how they could be converted into statements.

TABLE II. RULE BASE (NN –NOUN, PRP - PROPER NOUN, JJ – ADJECTIVE, MD – MODAL (CAN/ SHOULD/ WOULD/ COULD/ SHALL/ MAY/ MIGHT/ WILL))

New Rules introduced	
Has/Have/Had Verb PRP/NN Verb	Is/Are/Was there
Or Has/Have/Had PRP/NN Verb	Were there
How much/long PRP/NN MD/Verb	MD be there
How Verb	Is/Are/Was PRP/NN Verb
Is/Are/Was NN/PRP Verb	Were PRP/NN Verb
Were NN/PRP Verb	MD PRP/NN Verb
What/Which to Verb	Do/Does/Did NN/PRP Verb
When to Verb	When is/are/was PRP/NN Verb
Where to Verb	When were PRP/NN Verb
How to Verb	When is/are/was NN/JJ
What/Which NN/JJ	When were NN/JJ
What/Which has/have/had	When do/does/did

b) Triple Generation

In order to create SPARQL queries, we first generate RDF triples using the NL statements retrieved from the question to statement converter component. The RDF triples are basically Subject-Predicate-Object tuples of a NL statement. For generating RDF triples, we use open information extraction (OpenIE) systems that recognize relation phrases and associated arguments in arbitrary sentences without using any specific vocabulary. The main reason to use OpenIE systems was their ability to provide the triples required when building RDF triples without considering the domain.

Two OpenIE systems, Ollie [26] and Stanford OpenIE [27] were experimented with.

As an example, consider the following question: “Which hotels are the best in Colombo?”

- Question after POS-tagging: which/WDT hotels/NNS are/VBP the/DT best/JJS in/IN Colombo/NNP
- Statement : these hotels are the best in Colombo
- Triples generated from Ollie: these hotels;are the best in;Colombo
- Triples generated from Stanford OpenIE: hotels; are best in; Colombo

There were several instances where Ollie and Stanford OpenIE generated more than one tuple. During those situations, we chose the tuple with the highest confidence level. The confidence levels are auto generated with the triples by Ollie and OpenIE using a confidence function [20].

c) SPARQL Query Generation

For this component as well, a rule-based approach is used, where the RDF triples generated from the triple generation component and EAT generated from the question identification phase are used as inputs.

First the NEs and stop words in the triple are removed. Then EAT and the subject (first element of the RDF triple) are used to extract a relation list from the ontology. Moreover, using Subject and the Predicate (first and second elements of the RDF triple), a semantic similarity of the elements in the relation list is calculated. The element that has the highest semantic similarity in the relation list is taken as a parameter to check a rule base to identify the matching query pattern for the query template. Since the ontology was only completed with sights and hotels, we had to use query templates according to the data included in the ontology.

Consider things to do related query template[1] shown in Fig. 2,

```
PREFIX ontology: <http://www.*********/ontologies/ontology.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT ?name
WHERE {?place rdf:type ontology:NatureTrail. ?place ontology:displayName ?name}
```

Fig. 2. Query template for the question, "What good natural trails are in Sri Lanka"

By using the identified query template, a SPARQL query is generated. For an example, SPARQL query for the question "Which hotels are the best in Colombo?" is given below.

- EAT : Accommodation ; Hotels
- RDF Triple : these hotels;are the best in;Colombo
- Relation list extracted : Hotels
- SPARQL query generated is shown in Fig. 3[2]:

```
PREFIX ontology: <http://www.**********/ontologies/ontology.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT ?name
WHERE {?hotel rdf:type ontology:GotPool. ?hotel ontology:displayName ?name}
```

Fig. 3. SPARQL query for the question, "Which hotels are the best in Colombo?"

The SPARQL query obtained from the SPARQL generation component is used to traverse the ontology.

IV. Evaluation

A. Question Identification

Using a 500 question set for testing and a 4500 question set for training the model, classifier acquired a baseline (all questions classified under the most frequently observed class which is Travel Guide) accuracy of 24% for the coarse class and 9.7% for the fine class. Though all the combinations of the features were experimented, only the feature combinations shown in Table III gave good results. Our classifier outperformed the baseline classifier in all cases. It had a maximum accuracy of 0.87 for the coarse class and 0.76 accuracy for the fine class.

TABLE III. CLASSIFICATION ACCURACY FOR DIFFERENT FEATURE COMBINATIONS ON TWO LEVELS

Features	**Accuracy Coarse**	**Accuracy Fine**
ST(UG, BG)+POS(UG)+Synonyms of HW+WS	0.87	0.75
ST(UG, BG)+POS(UG)+Synonyms of HW+NE+ WS	0.86	0.76
ST(UG, BG)+POS(UG)+Synonyms of HW+WS+HW	0.85	0.74
ST(UG, BG)+Synonyms of HW	0.85	0.76
ST(BG)+POS(UG)+Synonyms of HW+WS	0.85	0.75
POS(UG)+Synonyms of HW	0.84	0.73
ST(BG)+POS(UG)+Synonyms of HW+WS+DBp	0.84	0.74
ST(UG, BG)+POS(UG)+Synonyms of HW+WS+DBp	0.84	0.74
POS(UG, BG)+Synonyms of HW+HW	0.83	0.73
HW+Synonyms of HW+NE+WS	0.56	0.41

Considering the results of Table III, it is understandable that "ST (UG, BG) + POS (UG) + Synonyms of HW + WS" gives the best accuracy for the coarse class, and fine class has the best accuracy for the feature combination of ST (UG, BG) + POS (UG) + Synonyms of HW + NE + WS. Moreover, it is evident that the question classifiers for travel related questions are more sensitive to the order of the words and the presence of the words.

In order to validate the manual annotation of the dataset, Kappa statistic was calculated. Four independent annotators were chosen without considering their knowledge on the domain and they were provided with the taxonomy and a question set of 100 to annotate. Kappa statistic for the coarse level annotation of the dataset resulted in 0.84, which emphasizes a near perfect agreement. The fine level annotation of the dataset resulted in 0.62, which shows a substantial agreement.

For other classifiers we experimented such as Naïve-Bayes, decision trees, and random forest classifiers showed accuracies of 0.48, 0.69 and 0.71 accuracies, respectively.

B. Searching the KB

The triple generation was validated with a dataset of 100 statements manually annotated with Subject-Predicate-Object triples. The triples generated from the OpenIEs for each of the statements were compared with the manually annotated dataset triples. During the comparison, Ollie resulted in an accuracy of 0.65, while Stanford OpenIE resulted in an accuracy of 0.6. Therefore it is evident that Ollie outperformed the Stanford OpenIE. Moreover, OpenIE consumes a large amount of memory for intermediate

[1] Changed due to proprietary issues
[2] Changed due to proprietary issues

garbage, which increases the computational intensiveness, thus making the system unresponsive sometimes.

V. Conclusion and Future Work

The results obtained for the question identification phase look promising though they are not up to the results of the state-of-the-art (In open domain, Huang et.al [12] recorded an accuracy of 93.4% for the coarse class and 89.2% for fine class). Ambiguities prevailing in the question base is recognized as the main cause for this state. The annotated question base and the taxonomy are available online [28] for public. The rule base approach did not show promising results due to the insufficient rule base.

As future work, we plan to use more domain-specific features such as gazetteer lists of hotels and attractions with the aid of other travel related ontologies and improve the syntactic head word extraction algorithm to extract less noisy head words and reduce the dimension of the vector space. We also plan to improve the rule bases we created during the second phase to achieve promising results covering a wide range.

Acknowledgment

We wish to thank the annotators who volunteered during the kappa evaluation.

References

[1] J. Allen, L. Schubert, G. Ferguson, P. Heeman, C. Hwang, T. Kato, M. Light, N. Martin, B. Miller, M. Poesio and D. Traum, "The TRAINS project: a case study in building a conversational planning agent", Journal of Experimental & Theoretical Artificial Intelligence, 1995 vol. 7, no. 1, pp. 7-48.

[2] S. Seneff and J. Polifroni, “Dialogue Management in the Mercury Flight Reservation System,” in Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems, 2000, vol. 3, pp. 11–16.

[3] Y. Yang, W.T. Yih and C. Meek, "WikiQA: A Challenge Dataset for Open-Domain Question Answering." in Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2015, pp. 2013-2018.

[4] E. Brill, S. Dumais and M. Banko, "An analysis of the AskMSR question-answering system." in Joint Conference on Empirical methods in natural language processing, Association for Computational Linguistics, 2002, pp. 257-264.

[5] A. Hallili, "Toward an ontology-based chatbot endowed with natural language processing and generation." in 26th European Summer School in Logic, Language & Information, 2014.

[6] X. Li and D. Roth, "Learning question classifiers." Proceedings of the 19th international conference on Computational linguistics. Association for Computational Linguistics, 2002, pp. 1-7.

[7] A. Chernov, V. Petukhova and D. Klakow, "Linguistically motivated question classification.", Proceedings of the 20th Nordic Conference of Computational Linguistics, Vilnius, 2015, no. 109, pp. 51-59.

[8] D. Zhang, and W.S. Lee, "Question classification using support vector machines." Proceedings of the 26th annual international ACM SIGIR conference on research and development in informaion retrieval, Association for Computational Linguistics 2003, pp. 26-32.

[9] K. Hacioglu, and W. Ward, "Question classification with support vector machines and error correcting codes." Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. Association for Computational Linguistics, 2003, pp. 28-30.

[10] M. Pota, M. Esposito and G. De Pietro, "A Forward-Selection Algorithm for SVM-Based Question Classification in Cognitive Systems." Intelligent Interactive Multimedia Systems and Services. Springer International Publishing, 2016, pp. 587-598.

[11] Y. Kim, "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882, 2014.

[12] Z. Huang, M. Thint and Z. Qin, "Question classification using head words and their hypernyms." Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2008, pp. 927-936.

[13] A. Andrenucci and E. Sneiders, "Automated question answering: review of the main approaches," Third International Conference on Information Technology and Applications, 2005, pp. 514-519.

[14] O. Ferret, B. Grau, M. Hurault-Plantet, G. Illouz and C. Jacquemin, "Document selection refinement based on linguistic features for QALC, a question answering system." Actes de RANLP,2001.

[15] G. Salton, "The SMART retrieval system—experiments in automatic document processing.", CA: Prentice-Hall, Inc. Upper Saddle River, NJ, USA,1971.

[16] R.D. Burke, K.J. Hammond, V. Kulyukin, S.L. Lytinen, N. Tomuro and S. Schoenberg, "Question answering from frequently asked question files: Experiences with the faq finder system." AI magazine, 18(2), 1997, p.57-67.

[17] C. Kwok, O. Etzioni and D.S. Weld, "Scaling question answering to the web." ACM Transactions on Information Systems (TOIS) 19.3 (2001),2001, pp.242-262.

[18] E. Sneiders, “Automated question answering: template-based approach”, Ph.D. dissertation, Data-och systemvetenskap, 2002.

[19] E. Sneiders, "Automated faq answering: Continued experience with shallow language understanding." Question Answering Systems. Papers from the 1999 AAAI Fall Symposium. 1999, pp. 97-107.

[20] M. Schmitz, R. Bart, S. Soderland and O. Etzioni, "Open language learning for information extraction." in Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics, 2012, pp. 523-534.

[21] Fernández-López, Mariano, "Overview of methodologies for building ontologies.", Proceedings of the International Joint Conference on Artificial Intelligence 1999

[22] J. Fu, S. Li and Y. Qu, "Question Matching Based on Domain Ontology and Description Logic," Second International Conference on Computer Research and Development, Kuala Lumpur, 2010, pp. 833-838.

[23] A. Fader, L. Zettlemoyer, and O. Etzioni, "Open question answering over curated and extracted knowledge bases.", Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. Association for Computational Linguistics, 2014. pp. 1156-1165.

[24] "TripAdvisor: Read Reviews, Compare Prices & Book", Tripadvisor.com,2017.[Online].Available:https://www.tripadvisor.co m/.

[25] "Lonely Planet", Lonelyplanet.com, 2017. [Online]. Available: http://www.lonelyplanet.com/.

[26] "Ollie - Open Information Extraction Software", Knowitall.github.io, 2017. [Online]. Available: https://knowitall.github.io/ollie/.

[27] The Stanford Natural Language Processing Group", Nlp.stanford.edu, 2017.[Online].Available:http://nlp.stanford.edu/software/openie.html.

[28] "LAPHR/Travel-Dataset-5000-", GitHub, 2017. [Online]. Available: https://github.com/LAPHR/Travel-Dataset-5000-.