Open-domain Question Answering Framework using Wikipedia

Saleem Ameen, Hyunsuk Chung, Soyeon Caren Han, Byeong Ho Kang

School of Engineering and ICT,
Tasmania 7005, Australia
{sameen, David.Chung, Soyeon.Han, Byeong.Kang}@utas.edu.au

Abstract. This paper explores the feasibility of implementing a model for an open domain, automated question and answering framework that leverages Wikipedia's knowledgebase. While Wikipedia implicitly comprises answers to common questions, the disambiguation of natural language and the difficulty of developing an information retrieval process that produces answers with specificity present pertinent challenges. However, observational analysis suggests that it is possible to discount the syntactical and lexical structure of a sentence in contexts where questions contain a specific target entity (words that identify a person, location or organisation) and that correspondingly query a property related to it. To investigate this, we implemented an algorithmic process that extracted the target entity from the question using CRF based named entity recognition (NER) and utilised all remaining words as *potential* properties. Using DBPedia, an ontological database of Wikipedia's knowledge, we searched for the closest matching property that would produce an answer by applying standardised string matching algorithms including the Levenshtein distance, similar text and Dice's coefficient. Our experimental results illustrate that using Wikipedia as a knowledgebase produces high precision for questions that contain a singular unambiguous entity as the subject, but lowered accuracy for questions where the entity exists as part of the object.

Keywords: open-domain, question answering, Wikipedia;

1 Introduction

The desire to produce a computational system that provides discrete answers to open ended questions with high precision and accuracy has been an area of academic inquiry for decades. While the advent of the World Wide Web and the prevalence of associated technologies such as search have presented an unmitigated opportunity for individuals to explore questions on dichotomous contexts, there remains pressing challenges – specificity, speed and relevance. There is a need for individuals to interact with information with greater ubiquity, efficiency and cohesion so that it reflects prevailing standards of social dogma. Question and answering (QA) systems have been a response to this problem but have operated with limited success in open domains. Despite the strides that research in natural language processing (NLP) has presented about the topological, ontological and semantic superstructures of expression; the problem of open-endedness has pertained due to the absence of a viable knowledgebase. Wikipedia, a perpetually expanding collation of information on diverse topics presents researchers with an opportunity to bridge the knowledge gap required to answer the dispersive distribution of questions presented within the open domain. This paper is therefore a preliminary investigation into the feasibility of utilising Wikipedia as a viable knowledgebase and through this study we explore the implications that this has for open-domain questions. One key tenet of Wikipedia that has strengthened its appeal is the prevalence of property-answer pairs for common characteristics related to people, locations and organisations. These properties can be extracted to present concise answers for factoid-based questions. We therefore place this as a constraint in our investigation, where we analyse entity specific questions, so that a constructive qualitative assessment about Wikipedia's knowledgebase is developed. Using named entity recognition (NER) to isolate the target entity in conjunction with standardised string matching algorithms such as the Levenshtein distance, similar text and Dice's coefficient, the efficacy of Wikipedia as the basis of a projection based question and answering framework is explored.

2 Related Works

There has been much research on Open-domain question answering (Open QA) for decades. Open QA systems require constructing a broad range of knowledge to achieve high accuracy [1]. Early systems focused on information retrieval approach [2]. Recently, large-scale knowledge base has been used to extract answer from a KB [3]. However, most of works require hand labeled on the logic that they applied to achieve high accuracy [4, 5]. A major research area in QA has been shifting to semantic parsing systems from small-sized and single-domain KBs to large-sized and multi-domain KBs [6-8]. Despite significant progress of constructing large-sized and multi-domain knowledgebase, major challenging problems of QA are still remained [9]. Recently, several QA systems proposed machine learning based systems that produce reasonable quality but still manual works are required such as designing lexicons, grammars and the knowledge base [4-6]. This paper investigates the feasibility of using large-sized multi-domain knowledgebase such as DBPedia.

3 Methodology

The objective of this paper was to investigate whether a framework that utilised Wikipedia's knowledgebase could be developed for an open domain question and answering system. We focused on examining the plausibility of answer retrieval for questions defined by two distinct characteristics: the existence of an entity and the prevalence of a related property. Our assumption was that if a question was comprised of a specific target entity, then that would be sufficient enough to query DBPedia's semantic knowledge graph by cross referencing the words embedded in the question against entity-specific keywords found within DBPedia using common string matching algorithms. This would enable an information retrieval process that could generate an answer to the user derived from the closest matching keyword. This process was implemented in stages using various methods: entity extraction using CRF based named entity recognition (NER); entity keyword generation using a SPARQL end-point to query DBPedia; property extrication using common string matching algorithms including the Levenshtein distance, similar text and Dice's coefficient; and answer representation through the implementation of a second SPARQL query using the entity and determined property. The processes implemented in this approach and the motivation for using it has been described in the following section.

3.1 Step 1: Entity Extraction – NER CRF Sequence Model.

In the context of NER, the observation sequence is defined by the token sequence of the sentence. Consequently, the probability of the label sequence can be based on the independent features of the observation sequence without the model having to account for the distribution of those dependencies, which enables greater diversity among the input features and therefore greater precision in entity recognition. Implementing the model requires a training process by credible domain-relevant sources such as a dictionary or WordNet. In this study, the CRF model used has been investigated by the Stanford NLP lab and was trained with the CoNLL English training data based on the characteristics of four distinct classes (entity types): person, location, organisation and misc. In this experiment, word sequences that were labelled as either a person, location or organisation, were defined as the target entity (the subject of the question) and words that were labelled as 'misc', were defined as potential properties that the user is specifically querying about the target entity.

3.2 Step 2: Keyword Generation – DBPedia and SPARQL.

The challenge associated with NLP within open domain question and answering is the underlying complexity of understanding the inherent meaning of the question that the user seeks to query. Even in situations where a singular subject target entity is successfully extracted using CRF-based NER, there remains the pressing problem of determining what the user query is actually asking about the target entity. Lexical and grammatical conventions suggest that it is reasonable to assume that questions marked by one entity subject, contains a distinct property that the user wants to ask about the entity within the remaining words of the question. This reduces the problem of understanding, to a simpler problem of property recognition. To put this into context, let us consider a scenario. Suppose a user wanted to ask 'Where was Barack Obama's place of birth?' NER would extract the target entity and subject of this question as 'Barack Obama', which places the remaining string of words 'where was place of birth' as potential candidate properties that the user wants to know about Barack Obama. Intuitively, it is clear that the property that the user wants to know about is Barack Obama's 'place of birth', however computationally determining this is difficult. In the aforementioned string, each word in isolation could be a property, each pair of words could be a property, or the entire string could be a property that the user wants information on and differentiation is a challenge. This is further exacerbated by the fact that there is no certainty that the answer to that property exists within Wikipedia's knowledgebase. The issue of property recognition is therefore twofold: 1. which word or word sequence should be selected as the target-entity property and 2. how do we determine whether the property selected has an answer?

Since the focus of our investigation was whether Wikipedia's knowledgebase alone was sufficient to answer questions that were characterised by one target entity, our SPARQL query extracted only the keyword properties that were defined by RDF data bound by Wikipedia's dataset and not the external RDF datasets linked to it. To contextualise this, suppose 'Barack Obama' was the target entity from the aforementioned example, DBPedia would return a list of all properties that were distinct to that entity based on its own algorithmic process that parses Wikipedia's dataset and labels each property accordingly. In our method, we extract all properties that are tagged with the RDF label 'dbp' or 'dbo', which represents property and ontology based characteristics respectively. The ontological based RDF label was introduced by DBPedia as a shallow, cross-domain ontology that was manually created to mitigate weaknesses in Wikipedia's infobox system related to data inconsistences, such as having different infoboxes for the same class, using different property names for the same property and not having clearly defined datatypes for property values (http://wiki.dbpedia.org/services-resources/ontology). DBPedia's ontology mappings produce a subsumption architecture of 685 classes that categorise 2,795 property types. Therefore, it was important to enact keyword generation that was based on both property and ontology labels, to produce a robust set of results that handled variation in the question asked by the user.

3.3 Step 3: Property Extraction – String Matching**.**

Using a string matching algorithm, we wanted to select the closest matching keyword derived from the DBPedia data – data that is known to have an answer – against the potential candidate properties defined from the NER process. The following table illustrates how the data is represented prior to the string matching process:

Original Question: Where was Barack Obama's place of birth?
Target Entity: Barack Obama

Potential Candidate Properties*	Keywords (from DBPedia)**
Where	Abstract
Was	Birth date
Place	Birth place
Of	Office
Birth	Party
	Predecessor
	Religion
	Residence
	Spouse
	…

The premise of string matching algorithms is to accommodate for the following scenario: provided two strings, a text and a pattern, determine whether the pattern exists in the text. In this study, the 'text' is defined as the list of properties derived from the question and the 'pattern' is marked by each of the keywords generated from DBPedia. Given the fact that we have a many:many relationship between these two lists, we approach the problem in the following way:

```
For each potential candidate property
 -   For each keyword
          o  Compare the candidate property string with the keyword
             string
```

```
        o   Store the percentage match in a variable
        o   If the percentage match of the current keyword is greater
            than the percentage match of the previous keyword, update
            the state of the variable for the closest matching key-
            word
  -   Store the closest matching keyword of each property in a proper-
      ty match array with its corresponding percentage value
For each value in the property match array
  -   Sequentially compare the percentage match of each value
  -   Select the closest matching property as the keyword sequence
      with the highest similarity measure
```

3.4 Step 4: Answer Extraction – using SPARQL and DBPedia.

The final step in our approach was to use both the extracted target entity and the keyword with the highest matching similarity, as the two elements required to query DBPedia for an answer using the RDF triple construct. Our goal was to select the distinct object for the entity, which had the property identified. The output from the SPARQL query produces the resulting object answer. Given the fact that the keywords used for string matching were based from properties that DBPedia had in its database about the target entity, our approach prevents the problem of using a property that cannot inherently be queried.

4 Evaluation

The goal of this paper is to explore the efficacy of using Wikipedia as a knowledgebase for an automated open ended question and answering system. Given the underlying complexity associated with existing question and answering architectures, which place a heavy emphasis on NLP, we wanted to investigate whether a basic algorithmic approach that utilised Wikipedia was sufficient to produce answers for common questions.

4.1 Data Collection

For our study, it was necessary to collect a series of random open ended questions to evaluate the efficacy of our approach. The Text REtrieval Conference (TREC), which focuses on research in information retrieval provides a track of sample questions for use in systems that retrieve answers for open-domain, closed-class questions that are primarily fact based. TREC provides six possible datasets ranging from the TREC-8 (1999) to TREC (2004). To evaluate this study, we opted to use the TREC-8 dataset. The motivation behind this choice was driven by the fact that the answers supplied for this dataset were definitive in their representation with an emphasis on singular answers. By using the TREC-8 dataset, we established the view that it would enable us to produce a more robust representation about the efficacy of our system.

One of the assumptions that we defined in this study was that we were specifically interested in the category of questions that contained a distinct target entity. However, in the dataset, not all questions were defined according to this quality. Therefore, any question that contained more than one unique target entity or that had a non-existent entity, according to our entity categories, were omitted from this study. Table 1 shows the distribution of the questions relative to characteristics associated with their entities. Of the 200 questions, 102 questions were used for our evaluation

Table 1. Distribution of data according to entity characteristics

Entity characteristics	Number of questions
Person Entities	29
Organisation Entities	14
Location Entities	59
Multiple Entities (Same Type)	14
Multiple Entities (Different Type)	16
No Entity Recorded	68

4.2 Named Entity Recognition

Given the fact that our ability to generate answers was dependent on the correct classification of the target entity, we first evaluated the success of the CRF-based NER in correctly identifying and labelling the entities present. Of the 102 questions, 100 questions had been labelled according to its correct entity type, which illustrates high precision in CRF-based NER, producing an average accuracy rate of 98.04%. While the CRF-based NER was successful in identifying the entity within the question, we note certain limitations in utilising the extracted entity to produce queries for all questions. A qualitative exploration of the data illustrates how the nature of the question in terms of its semantics, colloquialisms and linguistic structure has an impact on whether the identified entity can indeed be defined as the 'target entity' of the question. We discuss these observations according to the various entity types and comment on a characteristic that was found to be universal to all three.

Table 2. The distribution of entities that were flagged as presenting issues in being the 'target'

	Person	Location	Organisation
Name not specific	37.93%	18.64%	35.71%
Name exists in object	10.34%	23.73%	14.29%
Incorrect Entity Type	3.45%	1.69%	0.00%

4.2.1 NER for entity type 'person'

One of the problems associated with questions that were related to people was the divergent ways that a particular name could be represented as a result of status or specificity. Each of these will be discussed with examples.

Status: In this context, status defines the construct of a name, which is led by the title the individual holds within society, rather than the name they occupy. In popularised culture, certain people of influence are known according to a title such as 'President', 'Queen', which truncates the full name of the person, in favour of the popularised form used to identify them. For example, one of the questions in the TREC-8 dataset was "Who was President Cleveland's wife?". In popular culture, the target entity here should be identified as "President Cleveland", however CRF-based NER identifies the name of the individual exclusively, independent of their title and thus the name 'Cleveland' is labelled and extracted. While it correctly identifies the entity type as a 'person', querying DBPedia according to the name 'Cleveland' produces ambiguity, as Cleveland could refer to innumerable individuals, including the location of Cleveland. This problem was compounded by the variety of contexts that this pertained to such as "When was Queen Victoria born?" In this question, the target entity required is 'Queen Victoria', however since NER focuses on the name, the word "Victoria" is extracted instead.

Specificity: An extension to the problem of status is the problem of specificity. It was noted that as an individual's popularity is propelled in society, it is reasonable to identify them with a truncated version of their name. For example, a question such as "Where was Lincoln assassinated?" results in the entity 'Lincoln' extracted. While the semantics of the question enables human judgement to recognise that this question refers to the president Abraham Lincoln, a computer struggles to make the same association. In isolation, the term Lincoln can be associated with a multitude of identities, which makes disambiguation a problem. Despite the apparent magnitude of this problem, the embedded ontological mapping found in DBPedia appears to support names whose truncation is sufficient enough to remain unique. For instance, in the question "When did Nixon die?", the term 'Nixon' is redirected to the president 'Richard Nixon', which places limited liability on issues associated with specificity. Thus, even though 37.9% of the questions related to people had problems with specificity; leveraging the ontology mapping of DBPedia, significantly reduced the error factor to 13.8%.

4.2.2 NER for entity type 'location'

While the NER process was successful in identifying the target entity in several instances, it did not factor in the variation of syntactical and grammatical conventions where the entity identified was not necessarily the subject of the question, but in fact related to the object. To put this in context, consider the following question: "Which type of submarine was bought recently by South Korea?" Here, the subject of the question that is being asked about is the 'submarine' and the object that is related to it is 'South Korea'. NER in this case would extract the target entity as 'South Korea' and use it to direct the query. The issue with this, is that the properties related to South Korea are independent of the properties related to the submarine that was being queried about. This occurred 18.63% of the time, when analysing the cumulative distribution of the data. An interesting observation to note, was that 73.68% of these questions belonged to location based entities. We were unable to account for why the distribution skewed negatively towards location based entities, particularly given the uneven distribution of the sample size, however we were able to make speculative assertions. One speculation was the fact that it is more common to find questions that placed a location as a secondary component to a question as locations can often be used as descriptive terms to describe other objects, when compared to individuals or organisations.

4.2.3 NER for entity type 'organisation'

Organisation based NER featured the best performance among the various entity types. The main discrepancy identified was the same issue of specificity that marked 'people' based questions. Just like people, organisations that exhibited exuberant levels of popularity over the course of their organisational lifecycle had their names truncated in colloquial discourse, which necessitated a level of intuition to discriminate. For example, in the question “Where is the Bulls basketball team based?”, NER identifies the target entity correctly as ‘Bulls’. However, this is ineffective in querying DBPedia for keywords or answers as the word ‘bulls’ is not a unique identifier. Instead, the terminology ‘Chicago Bulls’ is a more apt description to produce the required results. This occurred 14.3% of the time, which closely aligns to our ‘people’ based analysis. Leveraging DBPedia’s ontology does not appear to be as important in organisation based queries as person based queries. One reason for this is because central to the identity of an organisation, is its name and thus, unlike people, it is rare for organisations to require the same level of semantic mapping.

4.3 Property Matching

Quintessential to the system’s ability to answer open-ended questions is the efficacy in which it identifies the target property within the question. Our approach involved querying DBPedia for a list of keywords related to the target entity and finding the keyword with the highest percentage similarity when compared to the terms within the question (excluding the entity). To evaluate its performance, we used three different string matching approaches (levenshstein distance, similar text and Dice’s coefficient) to determine whether there were any identifiable performance improvements that corresponded with the selected approach. Furthermore, we used three different approaches (full string, partial string, single word string) to segment the property list string within the original question to accommodate for scenarios where multiple words could define the composition of a particular keyword.

4.3.1 NER – universal issue: entity as the subject vs entity as the object

We first conducted a qualitative analysis of the property list that remained after the entity had been extracted from the original question to highlight distinguishing features. Our analysis primarily targeted the subset of questions that were independent of discrepancies derived in the entity selection process. The first observation we noted was that on average 38.9% of questions did not contain a clearly identifiable target property that was of relevance to the target entity; a necessary component to performing string comparison against a list of keywords.

Table 3. Sample Data that illustrates a subset of questions that contain no property

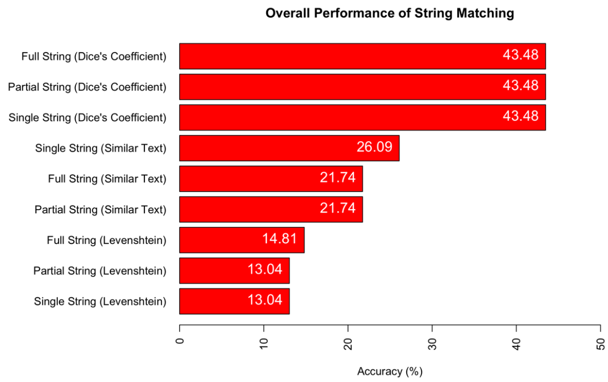
Original Question	**Target Entity**	**Property List**
Where is Qatar?	Qatar	Where is
Where is Dartmouth College?	Dartmouth College	Where is
What did Richard Feynman say upon hearing he would receive the Nobel Prize in Physics?	Richard Feynman	What did say upon hearing he would receive the Nobel Prize in Physics
In what year did Joe DiMaggio compile his 56-game hitting streak?	Joe DiMaggio	In what year did compile his 56-game hitting streak
Who came up with the name El Nino?	El Nino	Who came up with the name

The adverse implications of this can be best understood with reference to question one in Table 3, “Where is Qatar?” The target entity in this question is unmistakeably identifiable as ‘Qatar’, which enables our algorithmic process to produce a set of keywords related to it. This yields a resultant list of 110 potential keywords that can be utilised using the approaches described in section 4.3. The issue here is that for this to work, there must be a property that existed within the initial question that the keywords can liken a match to. However, the potential property strings that can be processed here are ‘where’, ‘is’, ‘where is’ – all of which are problematic in producing a reasonable match among the keyword list as they are not inherently the target property of the question, but instead are words that are reflective of ‘location’. In this case, each string matching technique produced entirely divergent matches. The levenshtein distance metric produced ‘timeZone’ as a match, followed by similar text and Dice’s coefficient that produced ‘metricFirst’ and ‘legislature’ respectively. This suggests that it is important to disregard matches that are below a particular threshold or alternatively, develop an approach that disambiguates key phrases by exploring a rule based approach. For instance, the word sequence ‘Where is’, which is often repeated among the questions, consistently references a location and can therefore be ontologically mapped to a more apt descriptive property. A second noteworthy observation pertains to questions that were compromised of multiple variables and therefore multiple properties. This subcategory of questions yields dispersive results as the key aspect of the question is chained in connection with a second element.

4.3.2 String Matching Approaches

For this evaluation, we wanted to determine whether using an algorithm that did not entail an underlying complex architecture would be sufficient in producing reasonable results. For this evaluation, we honed our analysis on the category of questions that aligned with our assumption and criteria. Out of the three string matching approaches that we implemented, as outlined in section 4.3, there were significant performance differences between the approaches. Dice’s coefficient illustrates that it produces stronger correlations between words of different form as shown in Figure 1.

Figure 1. Overall performance of string matching approaches in identifying the target keyword



4.3.3 Extending the Dataset: A Closer Examination

While the TREC-8 dataset was effective in providing a preliminary qualitative and quantitative evaluation of our approach, our findings suggested that the system was most effective in questions that aligned with specific criteria.

1. One specific target entity was identified
2. One relevant property related to the target entity existed
3. The property was related to the entity type as opposed to the entity itself.

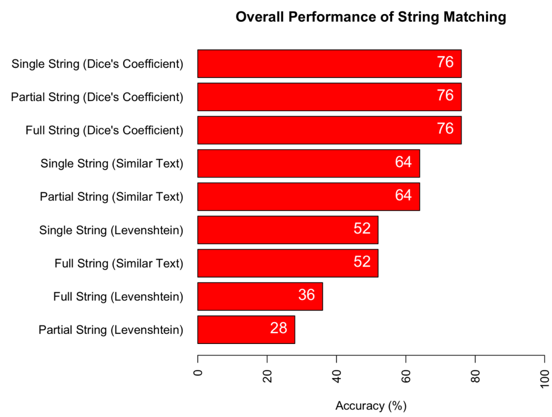
Criteria three is perhaps the most significant observation that was extrapolated. While DBPedia produces a robust set of results related to the entity, the keywords that it isolates have a tendency to be specific to the type of entity that the entity belongs to, as opposed to the specifics of the entity itself. To expound on this notion, consider general qualities that a user would relate to the entity type 'person'. Such qualities include, but are not limited to, 'birthday', 'religion', 'education, 'spouse', among several others. By the same token, consider the general qualities associated with a company such as 'founder', 'founding date', 'net worth', 'product' and more. These qualities are a subset of intrinsic characteristics related to the labelled entity's type as opposed to the explicit characteristics or experiences of the extracted entity. With this in mind, we were interested to investigate how the system performed against a dataset that aligned with these three qualities. The motivation behind this stemmed from the fact that by identifying a sub-category of questions that the system performed efficiently with, it would better enable us to extend the system so that it aligns with the rules associated with various categories of questions. Given the profound difficulty of finding a credible data source that aligned with the characteristics of questions that we were interested in investigating; it was necessary for us to develop a random question data distribution that could be utilised to explore the implications of this system. To maximise the variability of the data, we generated question templates for common questions related to people, organisations and locations.

Table 4. Sample Question Template

Where was X Y?	Which X is Y in?
Who is X Y?	In what year was X Y?
When is X Y?	When was X Y?
What is X Y?	Provide me with information regarding the Y of X.
How many Y does X have?	Who were the Y of X?
What Y does X have?	…

In Table 4, we document a series of question templates that accommodate for the various ways that basic questions that align with the stipulated criteria can be articulated. In these questions, X refers to the target entity and Y refers to the target property. Upon creating these question templates, we asked 50 students of diverse age and socio-ethnic backgrounds to produce a set of thirty questions distributed evenly between the respective entity type categories and place the name of a property that they deemed as relevant and related to the entity type in the position of Y as opposed to a property specific to the individual. While we considered randomly selecting properties from DBPedia's ontology network, we were interested in producing a dataset that was reflective of common questions that end users would be interested in querying about, guided by our question templates to ensure that our data considered different styles that questions could be asked. Upon collating the questions, we utilised NER and randomly selected a pool of 15 questions per category and re-evaluated the performance of the string matching algorithms.

Figure 2. Overall performance of string matching relative to guided question construction



We observe a similar pattern in the re-sampled dataset as the TREC-8 dataset, with an expected higher precision. It appears that the differentiator between the algorithm's effectiveness stems primarily from the string matching approach more than the segmentation technique applied. This is particularly evident in the utilisation of Dice's coefficient, as we achieve the exact same performance across all three segmentation categories, in the same way that was identified in the original TREC-8 dataset. The drastic performance improvement in comparison to the TREC-8 dataset stems from the fact that we isolated the question structure to align with the subset of questions that were observed to produce consistent results through our preliminary qualitative and quantitative evaluation. This affirms our postulation that Wikipedia's knowledgebase can offer a useful utility in answering questions that purport certain criteria – criteria that can be easily identifiable using basic algorithmic processes.

5 Conclusion

Ultimately, this study illustrates that Wikipedia is a viable knowledgebase for an automated open ended question and answering framework. A qualitative analysis indicated that there were a finite set of distinct problems that underpinned the basis of natural language such as identifying the subject versus the object of the question coupled with disambiguating keywords. It is clear that using NER in conjunction with a string matching algorithm in isolation is not sufficient enough to alleviate these problems. In the future, these problems can be mitigated by using proven approaches in NLP such as word sense disambiguation and semantic labelling which can help render guided queries within Wikipedia's knowledgebase. We found that by focusing on questions that contained three attributes – a singular target entity and a single property related to the entity type – yielded substantive performance improvements. This validates our preliminary investigation and suggests that Wikipedia's knowledgebase performs with efficacy within the domain of open-ended questions. By compounding Wikipedia's knowledgebase with more advanced NLP features that address the problems observed in our qualitative analysis of the TREC-8 dataset, we have confidence that Wikipedia plays a vital role in addressing the issue of open-endedness in future QA systems.

6 Acknowledgement

This work was supported by the Industrial Strategic Technology Development Program, 10052955, Experiential Knowledge Platform Development Research for the Acquisition and Utilization of Field Expert Knowledge, funded by the Ministry of Trade, Industry & Energy (MI, Korea). This work was supported as part of the the Office of Naval Research grant N62909-16-1-2219.

References

[1] Grosz, B.J., et al., *TEAM: an experiment in the design of transportable natural-language interfaces.* Artificial Intelligence, 1987. **32**(2): p. 173-243.

[2] Voorhees, E.M. and D.M. Tice. *Building a question answering test collection*. in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. 2000. ACM.

[3] Unger, C., et al. *Template-based question answering over RDF data*. in *Proceedings of the 21st international conference on World Wide Web*. 2012. ACM.

[4] Kwiatkowski, T., et al. *Scaling semantic parsers with on-the-fly ontology matching*. in *Association for Computational Linguistics (ACL)*. 2013.

[5] Berant, J., et al. *Semantic Parsing on Freebase from Question-Answer Pairs*. in *EMNLP*. 2013.

[6] Cai, Q. and A. Yates. *Large-scale Semantic Parsing via Schema Matching and Lexicon Extension*. in *ACL (1)*. 2013. Citeseer.

[7] Tsai, C., W.-t. Yih, and C. Burges, *Web-based question answering: Revisiting AskMSR*. 2015, Technical Report MSR-TR-2015-20, Microsoft Research.

[8] Liang, P., M.I. Jordan, and D. Klein, *Learning dependency-based compositional semantics.* Computational Linguistics, 2013. **39**(2): p. 389-446.

[9] Fader, A., L. Zettlemoyer, and O. Etzioni. *Open question answering over curated and extracted knowledge bases*. in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014. ACM.