CS563-QA: A Collection for Evaluating Question Answering Systems

Katerina Papantoniou[1,2] and Yannis Tzitzikas[1,2]

[1] *Computer Science Department, University of Crete, Greece*
[2] *Institute of Computer Science, FORTH-ICS, Greece*
{*kpapantoniou,tzitzik*}*@csd.uoc.gr*

July 4, 2019

Abstract

Question Answering (QA) is a challenging topic since it requires tackling the various difficulties of natural language understanding. Since evaluation is important not only for identifying the strong and weak points of the various techniques for QA, but also for facilitating the inception of new methods and techniques, in this paper we present a collection for evaluating QA methods over free text that we have created. Although it is a small collection, it contains cases of increasing difficulty, therefore it has an educational value and it can be used for rapid evaluation of QA systems.

1 Introduction

Question Answering (QA) systems aim at providing precise answers to questions, posed by users using natural language. Such systems are used in a wide range of application areas. QA is a challenging task since it requires tackling the various difficulties of natural language. Although the first QA systems were created long ago (back in 1960s), the problem is still open, the existing techniques have several limitations [4], and therefore QA is subject of continuous research. There is a wide range of techniques for QA ranging from simple regular expression-based methods, to methods relying on deep learning, and there are several survey papers including [2, 3, 6].

Since *evaluation* is important not only for identifying the strong and weak points of the various techniques (as well as their prerequisites), but also for facilitating the inception of new methods and techniques, in this paper we present a collection for evaluating QA methods. We focus on QA over plain text, i.e. the collection comprises free text.

This collection was constructed in the context of the graduate course CS563 of the Computer Science Department of the University of Crete (Spring 2019).

The rest of this paper is organized as follows Section 2 describes the objectives, Section 3 describes the collection, Section 4 briefly discusses metrics that can be used, Section 5 describes how the collection was used in the course and what the students achieved, and finally, Section 6 provides information about how to get the collection.

2 Objectives

Although there are several collections for evaluating QA systems (see [1]), the current collection has been constructed based on the following objectives:

- It should be small so that one can run experiments very *fast.*
- It should contain cases that require tackling *various kinds of difficulties.*
- It should contain cases of *increasing difficulty.*
- It should have *educational value.*

3 The Collection

The collection consists of *topics.* Each topic has an id, a title, a text (ranging from 40 to 18310 words), and a list of question-answer pairs. On average each topic has 5 question-answer pairs. The collection is represented in JSON (JavaScript Object Notation) format, an example is shown in Figure 1 in Appendix. In total the collection contains 149 question-answer pairs.

The topics are organized in groups of different levels of difficulty, starting from a group of relatively easy questions, ending up to a group with very hard to answer questions, or even impossible for the state-of-the-art methods. The groups are described in Table 1.

Group of Topics	**Difficulty**	**Indicative Tools/Resources**
1-12	Can be answered easily	Stanford CoreNLP
12-24	Lexical gap	Semantic dictionaries and ontologies, word embeddings (e.g. WordNet, Wiktionary, pretrained word embeddings)
25-41	Disambiguation, Inferences	world and commonsense knowledge (e.g. DBPedia Spotlight, DBPedia Ontology, YAGO)

Table 1: Groups of topics

3.1 The First Group

The *first group* (containing topics 1 to 12) consists of questions that do not require complex natural language processing to find the answers. The extraction of the answer is quite straightforward by applying basic linguistic analysis and by searching for simple patterns in the given snippets. This group includes questions that require tackling some form of noise (e.g. spelling errors) and possible the use of gazetteers. For example, indicative text-query-answer combinations of this category are shown in Table 2.

3.2 The Second Group

The *second group* of questions is composed by questions that require more complex linguistic manipulations in order to extract the correct answer. In this category, a lexical gap may be exist between the keywords in question and the content of the given snippet. Moreover, ambiguities may exist, e.g. POS ambiguity. The recognition of semantic relations is required in this group of questions that can be extracted with the help of semantic dictionaries, ontologies, word embeddings, etc. For example, indicative text-query-answer combinations of this category are shown in Table 3.

Topic	Wolfgang Amadeus Mozart
Text	Wolfgang Amadeus Mozart(27 January 1756 – 5 December 1791), to Leopold Mozart (1719 –1789) and Anna Maria, nee Pertl (1720 – 1778), was a prolific and influential composer of the classical era. Born in Salzburg, Mozart showed prodigious ability from his earliest childhood.
Question1:	Where was Mozart born?
Answer1:	Salzburg
Question3:	What was the first name of Mozart's father?
Answer3:	Leopold
Question4:	What was Mozart's profession?
Answer4:	composer

Table 2: Examples from Group A

Topic	Santiago de Compostela
Text	Santiago de Compostela is the capital of the autonomous community of Galicia, in northwestern Spain. The city has its origin in the shrine of Saint James the Great, now the Cathedral of Santiago de Compostela, as the destination of the Way of St. James, a leading Catholic pilgrimage route since the 9th century. In 1985, the city's Old Town was designated a UNESCO World Heritage Site. The population of the city in 2012 was 95,671 inhabitants, while the metropolitan area reaches 178,695. In 2010 there were 4,111 foreigners living in the city, representing 4.3% of the total population. The main nationalities are Brazilians (11%), Portuguese (8%) and Colombians (7%). By language, according to 2008 data, 21.17% of the population always speak in Galician, 15% always speak in Spanish, 31% mostly in Galician and the 32.17% mostly in Spanish. According to a Xunta de Galicia 2010 study the 38.% of the city primary and secondary education students had Galician as their mother tongue.
Question1:	How many people live in the city?
Answer1:	95,671
Question2:	What is the number of non-native Galician residents in Santiago de Compostela?
Answer2:	4,111

Table 3: Examples from Group B

3.3 The Third Group

The *last* and more complex group of questions consists of questions that require (a) some sort of ambiguity to be resolved, (b) world and commonsense knowledge, e.g. inferences that are based on knowledge found on resources beyond the given snippet or knowledge that all humans are expected to know. Cases of semantic and syntactic ambiguity, erroneous, partial or implied information and cases that refer not only to objective facts but also on sentimental opinions fall in this category. For these cases, a combination of the linguistic resources and tools of the previous category with world knowledge is required. Linguistic data and tools for the Linked Open Data Cloud such as DBPedia Ontology, YAGO and DBPedia Spotlight can be helpful is this group of questions. For example, indicative text-query-answer combinations of this category are shown in Table 4 and 5.

Text	The Low Countries, the Low Lands, is a coastal lowland region in northwestern Europe, forming the lower basin of the Rhine, Meuse, and Scheldt rivers, divided in the Middle Ages into numerous semi-independent principalities that consolidated in the countries of Belgium, Luxembourg, and the Netherlands, as well as today's French Flanders. Historically, the regions without access to the sea have linked themselves politically and economically to those with access to form various unions of ports and hinterland, stretching inland as far as parts of the German Rhineland. It is why that nowadays some parts of the Low Countries are actually hilly, like Luxembourg and the south of Belgium. Within the European Union the region's political grouping is still referred to as the Benelux (short for Belgium-Netherlands-Luxembourg).
Question1:	What countries belong to the Netherlands?
Answer1:	Belgium, Luxembourg, the Netherlands
Question2:	In which way Benelux countries are linked?
Answer2:	politically and economically

Table 4: Examples from Group C

Finally, this category also includes questions that require analyzing the pragmatics of the text for answering the questions (co-reference resolution, connections of sentences, speech acts, scripts, etc). For example, indicative text-query-answer combinations of this category are shown in Table 6

Text	As a freshman, he was a member of the Tar Heels' national championship team in 1982. Jordan joined the Bulls in 1984 as the third overall draft pick. He quickly emerged as a league star and entertained crowds with his prolific scoring. His leaping ability, demonstrated by performing slam dunks from the free throw line in Slam Dunk Contests, earned him the nicknames Air Jordan and His Airness. He also gained a reputation for being one of the best defensive players in basketball. In 1991, he won his first NBA championship with the Bulls, and followed that achievement with titles in 1992 and 1993, securing a ẗhree-peat.̈ Although Jordan abruptly retired from basketball before the beginning of the 1993–94 NBA season, and started a new career in Minor League Baseball, he returned to the Bulls in March 1995 and led them to three additional championships in 1996, 1997, and 1998, as well as a then-record 72 regular-season wins in the 1995–96 NBA season. Jordan retired for a second time in January 1999, but returned for two more NBA seasons from 2001 to 2003 as a member of the Wizards.
Question1:	What team did Michael Jordan play for **after** the Bulls?
Answer1:	Washington Wizards/Wizards

Table 5: Examples from Group C-temporal

Text	John likes Mary. He gives her a present.
Question1:	Who likes Mary?
Answer1:	John
Question2:	What Mary received?
Answer2:	A present
Text	John had a meeting in the university. However, he left for the university quite late this morning and he missed the bus of 8:30. When he arrived, Tom met him in the corridor, just a few meters from the entrance of the meeting room. He told him that everyone was waiting for him. He took a big breath and entered into the room with a big smile..
Question1:	Who smiled?
Answer1:	John
Question2:	How many were in the meeting room?
Answer2:	At least two
Question3:	What time John arrived at the meeting?
Answer3:	After 8:30
Text	John scored two goals in the last 5 minutes of the game giving 3 points to his football team. The goalkeeper of the other team was very sad.
Question1:	Why was the goalkeeper very sad?
Answer1:	Because he got 2 goals over the last 5 minutes of the game and his team lost.

Table 6: More Examples from Group C

3.4 What Phenomena are Covered by the Collection

Answer Types. The correct answers have been tagged with an *entity type*, i.e. it is the answer type in the context of question answering. The types of the answers of the questions in the collection are several, below we show them organized categories:

- *Time-related*: Year, Date, Time, Duration, Hour-related.
- *Agents*: Person, Organization, Team, Theatre, Company.
- *Roles*: Profession.
- *Locations*: Location, Location name, Region.
- *Artifacts, Materials, Ingredients*: Material, Ingredient, Object.
- *Quantitative*: Number, Percentage, Population, List.
- *Contact details*: Telephone, Address.
- *Monetary*: Money.
- *Terminology*: Medical Term, Genre.
- *Processes*: Recipe.
- *Misc*: Aspect, URL.

Question Types. The collection tries to cover a large number of question types to cover a wide range of users information needs. We use the *questions types* mentioned in [2]. Table 7 shows for each question type, one or more indicative topics of the collections that contain such a question (the list is not exhausting) and how many (approximately) such questions exist in the collection.

Question Type	Indicative Topics	Num of questions
Factoid	1,2,3, ...	> 60
Confirmation	3, 22, 27, ...	> 6
Definition	12, 28, ...	> 4
Causal	36	1
List	25, 32, 37	3
Opinionated	33	2
With Examples	37	2
Procedural	28	1
Comparative	36	2

Table 7: Coverage of Questions Types

Kinds of Difficulties. Moreover, Table 8 lists some phenomena/difficulties and some indicative topics of the collections where each phenomenon occurs and how many such questions (at minimum) exist in the collection.

4 Evaluation Metrics

We should note that creating the equivalent of a standard Information Retrieval test collection is a difficult problem. In a IR test collection, the unit that is judged, the document, has an unique identifier, and it is easy to decide whether a document retrieved is the same document that has been judged. For QA, the unit that is judged is the entire string that is returned by the system and quite often different QA runs return different answer strings, hence it is difficult to determine automatically whether the difference between a new string and a judged string is significant with respect to the correctness of the answer. One method to tackle this problem is to use so-called answer patterns and accept a string as correct if it matches an answer pattern

Difficulty	Topics	Num. of questions (at least)
morphological differences	13,14,16,17,...	32
wrong spelling of name	3, 36	2
syntax ambiguity	27, 29, 40	4
ambiguity of references	34, 35, 36	3
WSD	41, 42, 26	3
temporal reasoning	25,35,37,38	6
spatial reasoning	26	2
opinion sentiment reasoning	33, 36	5
comparison	36, 39	3
assumed script	40	2
assumed domain model	40	1
general historical knowledge	25, 37	4
speech act identification	-	0

Table 8: Coverage of Various Difficulties

for the question, answer patterns are described in [5]. In general, there are several methods and metric to evaluate QA systems, e.g. see [4]. Since we expect the answer of the system to be a single answer (not a ranked list of possible answers), set-based metrics are appropriate (i.e. not metrics for ranked results). Therefore for the collection at hand, we propose evaluating QA systems according to *accuracy.* If Q denotes the set of questions that are used in the evaluation, and AQC those that were answered correctly, then the Accuracy is the fraction of the questions that were answered correctly i.e. $Accuracy = \frac{|AQC|}{|Q|}$.

4.1 Disclaimer

Since the collection is small, any positive result cannot be straightforwardly generalized. This collection is useful as a first test in the sense that if a QA process does not behave well in this collection, then certainly it will not work in a bigger collection or in a real world case. If a QA process behaves well in this collection, then this does not necessarily mean that it will behave well in a bigger collection.

5 Experience in the Course

In the context of the course project of CS563, the students were given this collection and were asked to build a QA system. They were free to use whatever method, tool and external source they wanted to. Apart from the code of their system, the students had to evaluate their system.

5.1 Examination of the Projects

The delivered software should take as input any JSON file like the one of the collection. This enables the TA (Teaching Assistant) to use a slightly different JSON file during the grading/examination. Moreover, the students have to deliver a minimal User Interface (e.g. a console interface) enabling the user of the system (a) to select the text (by selecting one topic id from the JSON file), (b) to type a question in natural language, and (c) to view the response of the system. The format of response is free; apart from the short textual answer that is required for computing the metrics, students are free to provide responses having more complex form.

5.2 The Projects of Spring 2019

All projects focused on the first group of topics (1-12). Each student dedicated around 20 hours. The accuracy that they achieved ranges 16%-37% (i.e. 10 to 23 correctly answered questions from the 61). The approach that each project followed is summarized below.

Project Id	Project1
Process	Every topic is treated as a separate document corpus and each sentence in the topic's text as a document. Stemming and stopwords removal are applied to every sentence. The Okapi BM25 scoring function is used to rank sentences, while Stanford CoreNLP with a NER pipeline is used for detecting entities within the text. The ”wh-word” as well as other keywords in the question, are used for defining a set of relevant entity types. All entities of a relevant type are sorted based on the BM25 score of their sentence and the top scored entity is the answer that is returned by the system.
Components:	CoreNLP
Time spent:	20 hours
Accuracy	20/61= 32%

Project Id	Project2
Process	This project is based on Named Entity Recognition, stemming (Porter's stemmer) and Jaccard similarity. The entity type of the sought answer is extracted by simple patterns, i.e. by checking if the first word of the question is Who, When, Where, etc. All keywords in the text whose entity type is the same as the type of the sought answer, are considered as candidate answers. These candidates are scored based on the Jaccard similarity between the sentence they occur and the question. If a candidate appears in more than one sentence, then the maximum score is kept. Before calculating the similarity, stopwords are removed from the sentences and also stemming is performed.
Components:	CoreNLP
Time spent:	20 hours
Accuracy	23/61 = 37%

Project Id	Project3
Process	Given a question in natural language and the searched entity type, this project indexes at first the topics of the collection (removed stopwords, stemmed terms) and afterwards the sentences of that topic in order to find the most relevant one (by calculating the cosine similarity between the query and each topic/sentence). Following, it analyzes the tokens of the retrieved sentence (using coreNLP and user-defined regular expressions) and returns the questioned entity. In general, the application was based on basic linguistic analysis and searching of simple patterns in the given snippets. Furthermore, “Wiktionary”, a semantic dictionary, was used for the recognition of semantic relations. In order to premium the primary tokens, the “tf” of the related ones (hyponyms, synonyms etc.) was reduced by 20 per cent.
Components:	CoreNLP, Wictionary
Time spent:	5 days * 8 hours = 40 hours (approx)
Accuracy	10/61 = 16%.

Project Id	Project4
Process	This project focused on factoid questions. It uses ElasticSearch[1] for building an index out of every topic. Each sentence is indexed as a document with multiple fields. Using NER from CoreNLP, a named-enity field holds the corresponding information that is extracted. An additional field holds the coreference information. At answer time the best sentence is returned, using a boolean-multimatch ES query, and a filter is applied for retrieving only the relevant documents (sentences) that also contain as a field the same named-entity type with the question. If more than one answers of the same type are identified, the system splits sentences into phrases and returns the one that is closest to the matching query keywords.
Components:	ElasticSearch, CoreNLP for NER & Coref
Time spent:	25 hours
Accuracy	18/61 = 29%

6 How to Get the Collection

The collection is public and can be downloaded from the Open Data Catalog of ISL[2]. It is a JSON file structured as shown in Figure 1. The license of the catalog is: non-commercial use, attribution.

Acknowledgement

Many thanks to the students that contributed to this report including Giorgos Kadilierakis, Kostas Manioudakis, Maria-Evangelia Papadaki and Michalis Vardoulakis.

[2] http://islcatalog.ics.forth.gr/dataset/cs563-qa-an-evaluation-collection-for-question-answering

References

[1] E. Dimitrakis, K. Sgontzos, and Y. Tzitzikas. A survey on question answering systems over linked data and documents, 2019. (under submission).

[2] Amit Mishra and Sanjay Kumar Jain. A survey on question answering systems with classification. *Journal of King Saud University-Computer and Information Sciences*, 28(3):345–361, 2016.

[3] Barun Patra. A survey of community question answering. *CoRR*, abs/1705.04009, 2017.

[4] Alvaro Rodrigo and Anselmo Peñas. A study about the future evaluation of question-answering systems. *Knowledge-Based Systems*, 137:83–93, 2017.

[5] Ellen M Voorhees and Dawn 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*, pages 200–207. ACM, 2000.

[6] Mengqiu Wang. A survey of answer extraction techniques in factoid question answering. *Computational Linguistics*, 1(1), 2006.

A Example of a Topic

An example of a topic is given in Figure 1.

```
{
  "id": 7,
  "title": "Tiramisu",
  "text": "Tiramisu (from the Italian language, spelled tiramisu [?tirami?su],
  meaning ’pick me up’ or ’cheer me up’) is a coffee-flavoured Italian dessert.
  It is made of ladyfingers (savoiardi) dipped in coffee, layered with a
  whipped mixture of eggs, sugar, and mascarpone cheese, flavoured with cocoa.
  The recipe has been adapted into many varieties of cakes and other desserts.
  Its origins are often disputed among Italian regions of Veneto and
  Friuli-Venezia Giulia.   Most accounts of the origin of tiramisu date its
  invention to the 1960s in the region of Veneto, Italy, at the restaurant
  ’Le Beccherie’ in Treviso. Specifically, the dish is claimed to have first been
  created by a confectioner named Roberto Linguanotto, owner of ’Le Beccherie’.
  Category  Dessert recipes Servings 4-6 Energy  200 Cal (800 kJ)
  Time 15 minutes + 1 hour refrigeration
  Difficulty: 2/5   Ingredients for 6{8 people  4 egg whites  2 egg yolks 100 g
  (1/2 cup) of sugar  500 g (2 1/2 cups) of mascarpone cheese 4 small coffee
  cups of espresso coffee 400 g of lady fingers (savoiardi) (or sponge cake)
  unsweetened cocoa powder to sprinkle   over before serving
  Preparation Make espresso coffee, let it cool a bit.
  Separate the egg yolks and the whites of two eggs in two bowls.
  Beat sugar into the egg yolks.
  Beat the Mascarpone into the sweetened yolks.
  Add two more egg whites to the other two and whisk  until they form stiff peaks.
  Fold gently egg whites into Mascarpone mixture.
  Quickly dip both sides of the ladyfingers in the espresso.
  Layer soaked ladyfingers and Mascarpone in a  large bowl or pan
  (start with fingers, finish with mascarpone). Sprinkle
  unsweetened cocoa powder on top just before to serve.  Refrigerate for one hour or two.",
  "qa": [
    {
      "question": "What tiramisu means?",
      "answer": "pick me up / cheer me up",
      "entity": "UNKNOWN",
      "note": "group A, factoid"
    },
    {
      "question": "From which Italian regions has its origins this dessert?",
      "answer": "Veneto and Friuli-Venezia Giulia",
      "entity": "LOCATION",
      "note": "group A, factoid"
    },
    {
      "question": "What is the name of the restaurant that the first tiramisu was made?",
      "answer": "Le Beccherie",
      "entity": "ORGANIZATION",
      "note": "group A, factoid"
    },
    {
      "question": "Who has made the first tiramisu?",
      "answer": "Roberto Linguanotto",
      "entity": "PERSON",
      "note": "group A, factoid"
    },
    {
      "question": "What kind of coffee is used in tiramisu?",
      "answer": "espresso",
      "entity": "INGREDIENT",
      "note": "group A, factoid"
    },
    {
      "question": "List some ingredients of tiramisu?",
      "answer": "eggs, sugar, mascarpone cheese, espresso coffee, cocoa",
      "entity": "INGREDIENT",
      "note": "group A, list"
    }
  ]
},
```

Figure 1: An example of a topic