ÇANKAYA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

Project Report

CENG 408

Innovative System Design and Development II

P2017-8
Search by Question

Mehmet Şerefoğlu- 201611688
Koray Danışma- 201511409
Ali Dorukhan Karakaya- 201311027

Advisor: *Dr. Engin Demir*

Table of Contents

List of Figures

1.Introduction

1.1 Purpose

The purpose of this document is describing Search by Question: Find results of asked 5w1h question in the document collection and list them. It aims to access an information faster even almost all category; history, science, sport, education etc. Moreover, it will ve explained to algorithms of how the system find answers of questions.

Today's wealth of big data and the need for more complex evidence based decisions such as fails to keep up with available information cognitive computing enables people to create a profoundly new kind of value finding answers and insights locked away in volumes of data whether we consider a doctor diagnosing a patient a wealth manager advising a client on their retirement portfolio or even a chef creating a new recipe they need new approaches to put into context the volume of information they deal with on daily basis in order to derive value from it this process serves to enhance human expertise.

Search by question system produces answers of asked question in English. Investigating about system started in 1960s . Early search by question system, is called question answering system that is basis of search by question, were developed for limited capabilities. In this project, system focus on kinds of questions which is asked by users, characteristics of refered data sources, and structures of produced correct answers. For identifying the future scope of search by question, this document includes also analyzes of system since its origin and classifies them based on varies criteria.

1.2. Overview of the Project

Search by Question is basis of computers that will try to answer textual questions. The main point here is natural language processing, NLP for short, is a method for computers to analyze, understand, and make meaning from human language in a smart way. Through utilizing NLP, developers can promote and structure knowledge to perform works such translation, automatic summarization, named entity recognition, sentiment analysis, relationship extraction, speech recognition, topic segmentation etc. NLP at the present is based on deep learning. This models needs large amounts of taged data to learn and specify relevant relations, and combining this type of big data set is one of the main aims to NLP.

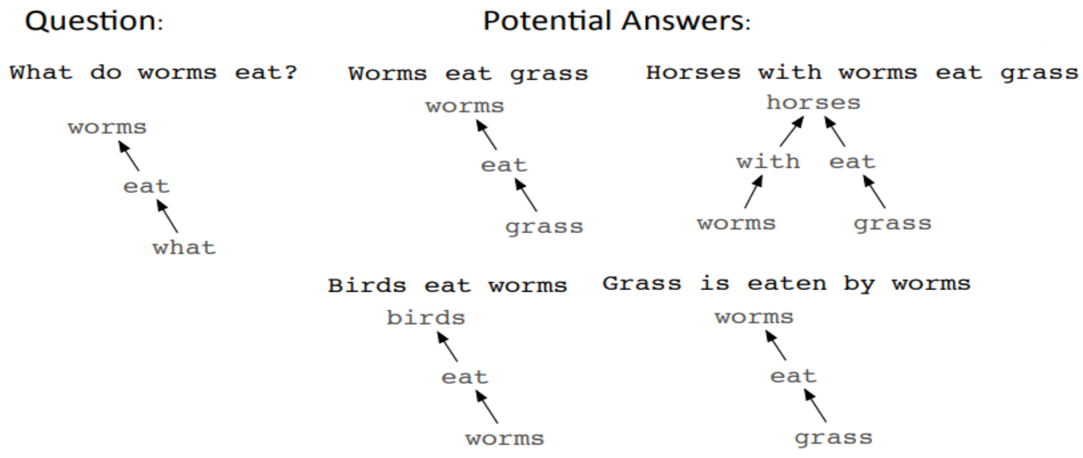

Figure-1: One of the earlier NLP tasks (In 1960s)

To program previous NLPs, worked on a more rule based solution which NLP just notices what words and phrases to search for in question to give particular answers when these phrases appeared again. This solution in process with machine learning algorithms. On the other hand, deep learning is more flexible. Because intuitive approach in that algorithms learn to specify intent of user from a lot of examples.

Basis of Search by Question is also question answering. For systems implementing, there were two significant process of question answering: IR-based question answering and knowledgebased question answering.

Test about question answering was the computer that worked on answer the question that was “the Great Question Of Life The Universe and Everything” . Computer’s answer was 42, but the details of the this asked question were never unclosed. This took place in TV program that is called the Hitchhiker's Guide to the Galaxy. Another test was IBM’s Watson question-answering system, which is used for this project, won the game is called Jeopardy. The system answer the following question: What is your most inspired Novel the Author’s of "An account of the principalities of Wallachia and Moldovia”. The system answer both faster and logical to this question then human.

Search by Question more focuses on factoid questions. That is one of the significant difference between Question Answering and Search by Question. Factoid questions can be answered with basic facts expressed in any short text answer. For example, the following factoid questions can be answered with a short string expressing a date, personal name, or location:

(i) When was Ataturk born?
(ii) Who is the founder of Microsoft?
(iii) Where is Başar Soft based?

The expected answers depend on kinds of asked questions. System deals with different kinds of questions and require different strategies to detect answers. Thus, following classifies questions on basis of kinds of asked question: factoid (what, where, when, why, who, how), confirmation (is, was, will etc.), hypothetical (what would happen), and also causal (how or why)

Analysis of one of the significant processes that is used for Search by Question is in the below, with focusing on its application to factoid questions.

2. Overall Description

2.1. Search by Question with IR-based Factoid Questions

The aim of IR-based search by question is finding short text parts on the document sets for answer a question. There are three states of Search by Question with IR-based Factoid Questions: question processing, passage retrieval, answer processing.

2.1.1. Question Processing

The purpose of the question processing state is extracting a number of components of information from the question. The answer type defines the type of entity the answer consists of (person, date, location, etc.). The query defines the keywords that is used for IR system for using in searching for documents. Some systems extract a focus, that is the string of words in the question that are replaced in any answer string found by the answer. Some systems also categorize the question kind that could be a definition question, a y/n question, a math question. For example, for the following question:

Which universities have computer engineering departments in Turkey?

The process of query produce results as the following:

Answer Type: university

Query: universities, computer engineering department, in Turkey

Focus: computer engineering department

There are two most commonly tasks for question processing. Question classification and query formulation.

2.1.1.1. Question Classification

Question classification task is used to specify kind of answer and similar class classifying of answer or named entity. Question as “Who leads Journey to Mars Project of NASA?” expects answer kind of person or question as “Where is the hagia sophia museum?” expects answer kind of location. In addition to, if user know the answer kind, every sentences or noun phrases in all documents for answer should not be analyzed by user, instead looking at, for example, just person.

As analyzes so far in this document, user could arrange set of probable answer kinds for question classifier from named entities as location, date, person etc. On the other hand, the most commonly used set of answer kind is answer type taxonomy. This kind of taxonomies can be composed semi-automatically and dynamically, for example, WordNet[1] ,or user can design them by manually.

The reason of why WordNet is needed is some problems that might be occured with named entity. Because system with named entity suppose that every answers are named entitiy. This situation oversimplifies the realazing question and also language. Analyzes of following examples could make the status clearer:

Which graphic cards can run x game?

Here system perceives the "graphic card" term as a named entitiy.

Who indites x article?

There is no guarantee for matching "indite" and "write". Because relation between alternations of question and answer expressions frequently not documented.

For the answer to first example, which grapgic cards can run x game?, system need to recognize that the answer is a type of graphic card and meaning of the word graphic card. To do this, WordNet encodes all synonyms of graphic card word, graphic card as a use for computer systems and then, nouns that from recalling passages could be investigated against synonyms.

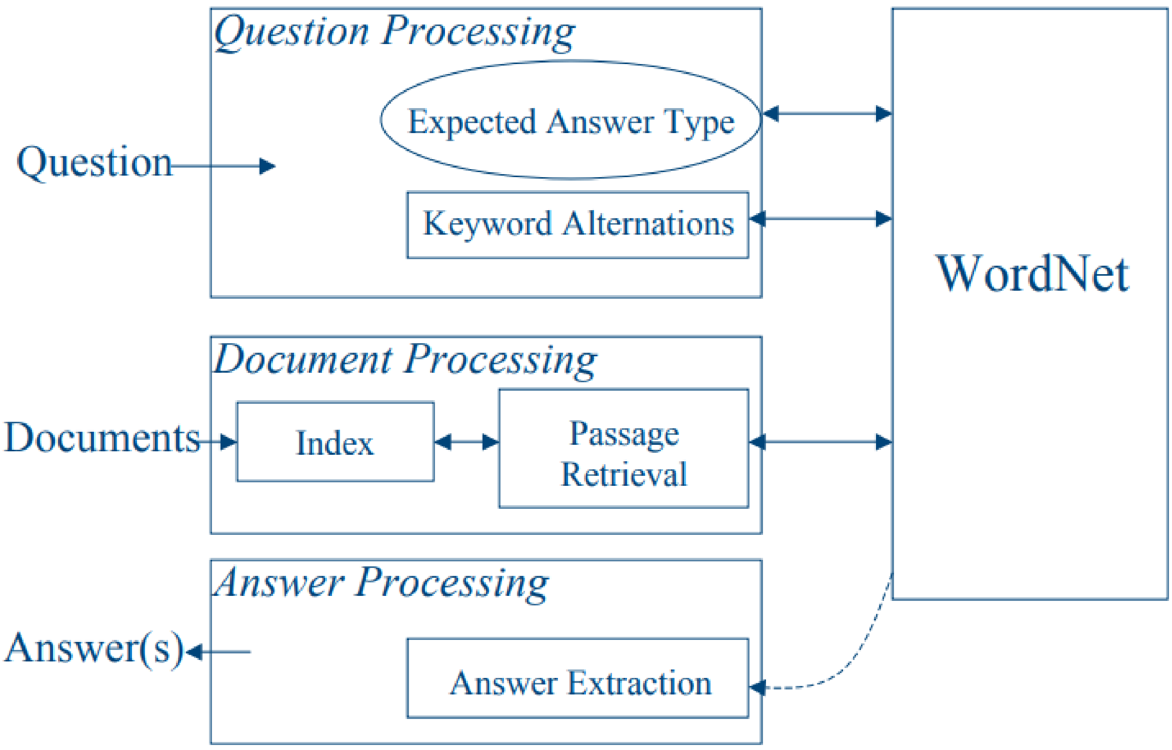

Figure-2: Interaction between WordNet and Search by Question System

Question classifiers could also be designed by hand writing rules, by managed machine learning, or with some combination.For example, The Webclopedia QA Typolog includes 276 hand written rules related with the almost 180 answer kinds in the typology[2]. Regular expression rule for defening an answer kind like definition, which supposes that asked question has been named entity labelled, could be

what {is | was | are | were} term(s)

More advanced question classifiers are based on managed machine learning, and worked on databases of questions which have been tagged manually with an answer kind[3]. Typical properties which used for classification contain words in the questions, the part of speech of each word, and also named entities in the questions.

Designing question classifiers with managed machine learning helps to use semantic information related with words in asked questions. The WordNet sync ID of word could be used as a property, as the IDs of hypernym and hyponyms of each word in asked question. In summary, precision of question classification is high on simple question kinds like location, date, and person questions. Specifying reason and description questions could be harder.

2.1.1.2. Query Formulation

Query formulation task is to create a list of keywords, which form a query that can be sent to information retrieval system, from asked question. What query to create also depends on application. If system is applied to Internet, keyword that is from every word in asked question can be created, and allow web search engine to remove any stopwords. Generally, the question word (where, when, etc.) is leaved out. In addition to, keywords could be created from just the expressions situated in the noun phrases in asked question, applying stopword lists to ignore verbs that mostly existing in searching text, also verbs with low content and functionally words.

Class	#	Class	#
ABBREVIATION	18	term	19
abbreviation	2	vehicle	7
expression	16	word	0
DESCRIPTION	153	**HUMAN**	171
definition	126	group	24
description	13	individual	140
manner	7	title	4
reason	7	description	3
ENTITY	174	**LOCATION**	195
animal	27	city	44
body	5	country	21
color	12	mountain	5
creative	14	other	114
currency	8	state	11
disease/medicine	3	**NUMERIC**	289
event	6	code	1
food	7	count	22
instrument	1	date	146
lang	3	distance	38
letter	0	money	9
other	19	order	0
plant	7	other	24
product	9	period	18
religion	1	percent	7
sport	3	speed	9
substance	20	temp	7
symbol	2	vol.size	4
technique	1	weight	4

Figure-3: Sample question typology

For the table in Figure-3[4], asked question could be tagged either with a dark tag like location or description or with other tag like location:country, description:reason. # is also number of questions in each class. The questions are manually classified by user.

A query formulation works for implementing query reformulation rules to the query. This working is also used for questioning the Web. The rules reinvestigate asked question to query reformulation for making it substring of probable answers. For example, the question "when was computer invented?" is reformulated as "the computer was invented". Here there is examples for hand written reformulation rules:[5]

When was A invented -> A was invented ?x

?x represents location of the expected answer.

2.1.2. Passage Retrieval

The query which was created in the question processing state is next used to query an information retrieval system. It is either a general IR engine on a exclusive set of indexed documents or a Web search engine. The result of this document retrieval phase is a set of documents.

Even though set of documents is usually ranked by relevance, the answer to the question is probably not top ranked document. Because documents are not an convenient unit to rank with respect to the purpose of a search by question system. A highly relevant and large document that does not distinctly answer a question is not an optimal candidate for further processing.

Hence, the next state is extracting a set of possible answer passages from the retrieved set of documents. The description of a passage is necessarily system dependent, but the characteristic units contain paragraphs, sentences, and sections.

After these processes, passage retrieval could be performed. First, passages that in passage retrieval the returned documents which don't include possible answers filtered out and then rank the rest by similarity of them to include an answer to the question. The first stage in this process is runing a named entity or answer kind classification on the retrieved passages. The answer kind which is determined from the question specify that the potential answer kinds that are expected to see in the answer. Therefore, documents which don't include any entities of the right kind can be filtered out.

The remaining passages are ranked, generally by supervised machine learning, trusting a small set of properties that could be smoothly extracted from a probable large number of answer passages, like: The rank of the document from that passage was extracted, the count of question keywords in the passage, the longest exact set of question keywords that takes place in the passage, the count of named entities of right kind in the passage. The following figure show this step with pseudocode.

```
(1) retrieve passages using keyword
  set K and proximity p.
(2) if number of passages < MinPass:
      if p < MaxProx
           increment p; goto step (1)
      else
           reset p;
           drop the least-significant keyword from K;
           goto step (1)
(3) else if number of passages > MaxPass:
      if p > MinProx
           decrement p; goto step (1)
      else
           reset p; add the next available keyword to K;
           goto step (1)
(4) return the current set of passages.
```

Figure-4: pseudocode example of part of the passage retrieval

2.1.3. Answer Processing

The final state of search by question with IR-based factoid questions is extracting a particular answer from the passage. So, the user with an answer like 4,2 light years to the question “How many light years there are between Earth and Proxima b?” could be presented.

Two classes of algorithms have been implemented to the answer processing task, one of them based on answer type pattern extraction and another one based on N-gram tiling.

In the answer type pattern extraction methods for answer processing, information about the expected answer type together with regular expression forms is used. For example, for questions that’s answer type is a music , the answer type or entity that is named tagger on the candidate passage or sentence and return which entity is tagged with type music are run. Thus, examples that are in the below, the entities which are bold named are extracted from the candidate answer passages for the answer to the music and location questions:

“What is the name of Tarkan's last album?”

In September Tarkan’s last album **10** will be on the market.

“What is the capital of Turkey?”

This international conference will be take place in **Ankara**, capital of Turkey.

On the other hand, the answers to some questions, like definition questions, are not similar to be a spesific named entity type. For these type of questions, instead of using answer types, hand written regular expression forms are used to support for extracting the answer. Moreover, these forms are useful in situations in that a passage includes multiple examples of the equal named entity type. Following examples show some forms for the question phrase (QP) and answer phrase (AP) of definition questions:

Form: <AP> such as <QP>

Question: What is react native?

Answer: Developers use this type of tool that used for building native mobile apps with JavaScript such as react native

or

Form: <QP> ,a <AP>

Question: What is blimp?

Answer: Because of defecting of blimp, a housing attached to a camera which reduces the sound caused by the shutter click, plans are postponed.

The forms are particular to each question kind and are either written by manually or learned automatically by using relation extraction methods. Forms could be used with other information as properties in a classifier which ranks candidate answers.

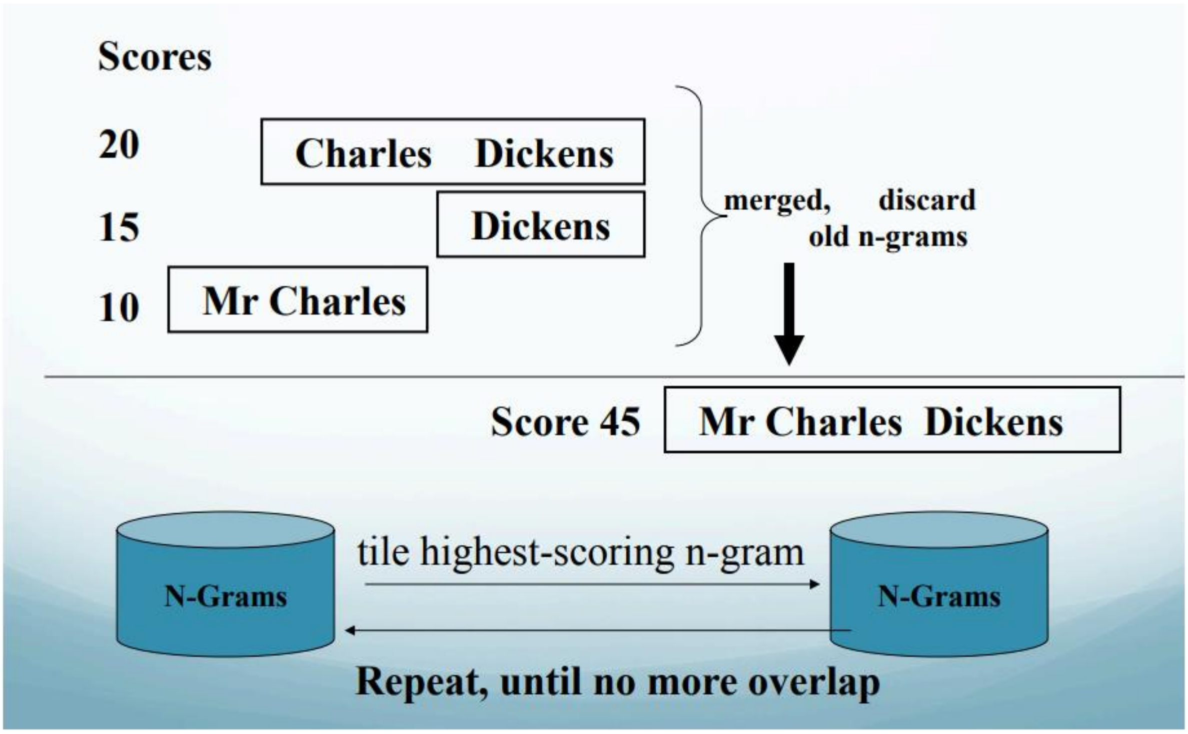

Figure-5: Process of N-gram Tilling

There is a another way to answer extraction, use Web search that is based on N-gram tiling, or its other name is redundancy based approach[6]. First step of this method is that snippets returned from the Web search engine, generated by a formulated query. Then, N-gram mining, every unigram, bigram, and trigram taking place in the snippet is extracted and weighted. The weight is a function of the count of snippets in that the N-gram taking place, and the weight of the query formulation forms that returned it. In the N-gram filtering state, N-grams are ranked by relevancing they match the expected answer type. Scores which are result of ranked are sum of weights of the query that retrieved coincident pages that build for each answer type. Then, canditates forms should be found. This occurs that for a given question kind, detect an question with question term and answer term, submit to search engine, download top X web documents, select relevant sentences for answer term, detect all substring and their numbers, select states with question term and answer term and replace them. The following example shows N-tilling process:

Question: When was Alan Turing Born?

Answer: Alan Turing (1912)

Question Term: Alan Turing

Answer Term: 1912

- Alan Turing (1912-1954) was highly influential in the development of theoretical computer science
- Alan Turing (1912-1954) devised a number of techniques for speeding the breaking of enemy ciphers during second world war.

Phrase: Alan Turing (1912-1954), count= 2

Convert to: <Name> (<Answer>)

2.2. Using Multiple Information Sources: IBM's Watson

There are 4 stages of the DeepQA system, question-answer system of Watson, that is the search by question component of Watson.

The first state is question processing. The DeepQA system runs investigating the grammers, named entity tagging, and correlation extraction on the question. Then, such as the text based systems in Section 2.1, the DeepQA system extracts the focus, the answer kind, other name is the lexical answer type or LAT, and implement question classification and question sectioning.

After that, DeepQA extracts the question focus. The focus is the component of the question that co refers with the answer, and it is used to arrange with a supporting passage. Hand written rules extract the focus. This process is such as a rule extracting for any noun phrase with adverb "this", and rules extracting pronouns that are such as he, she, him, her.

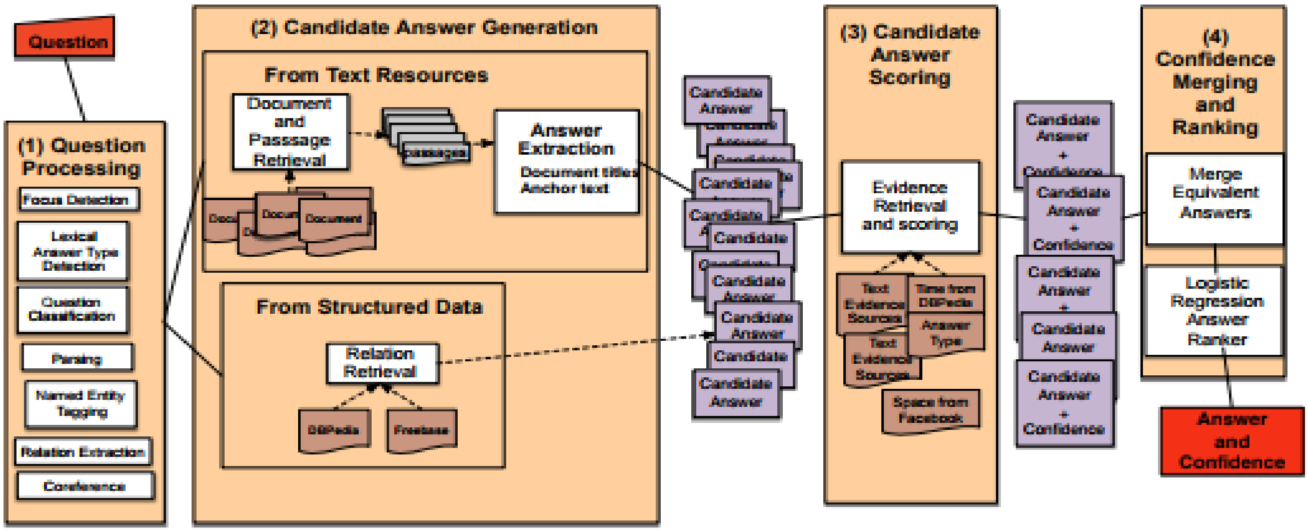

Figure-6: The 4 general states of Watson

The lexical answer type is a one word or words that explain lexical answer type about the semantic kind of answer. Lexical answer types are also extracted by rules: the default rule is to select the syntactic headword of focus. Other rules evolve this default selection. For example extra lexical answer types can be words in question which are have a specific syntactic relation with the focus, like headwords of predicative nominatives of focus. In some cases, category can proceed as a lexical answer type, if it refers to a type of entity that is appropriate with the other lexical answer types. Additionally, using the rules straight as classifier, they could instead be used to return a possibility as well as a lexical answer type.

Difference between DeepQA and IR-based factoid question answerers is that in IR-based question answerers, first answer type is determined, then using a strict filtering algorithm for taked notice text strings which have that type. On the contrary, In DeepQA, lots of answers are extracted, determine a set of answer types, then turn of 'candidate answer scoring' state, finally, simply score how well each answer appropriate the answer types to be one of many sources of proof.

In the second state, the operated question with external documents are combined. These candidate answers are extracted from text documents as section 2.1, first generate a query from the question, for DeepQA this is usually completed by sifting stop words, then upweighting any terms that take place in any correlation with focus.

The third state is that use a lot of sources of proof to score the candidates. One of the most significant is the lexical answer type. DeepQA contains a system which attracts a candidate answer and a lexical answer type and returns a score representing whether the candidate answer could be interpreted as subclass or model of the answer type. For example, the candidate "chronic headache" and the lexical answer type "permanent illness". DeepQA matches these words one by one with probable entities in internal medicine on web sites like DBpedia and Wikipedia. Hence, the candidate "chronic headache" is matched with the DBpedia entity "migraine", then that model is designed to the Wikipedia kind "symptom". The answer type "permanent illness" is designed to the Wikipedia kind "indication".

In the final state, which is answer merging and scoring stage, first candidate answers that are equivalent are merged. Hence, if two candidate answers are extracted like A. Turing and Alan Turing, this state would merge two into a single candidate. For proper nouns, automatically produced name dictionaries could support this task. One useful type of resource is large synonym dictionaries which are composed by listing all anchor text strings which point out to same Wikipedia page; this kind of dictionaries give large numbers of synonyms for each Wikipedia title, for example, A. Turing, Alan Turing, Computer Scientist Alan Turing , Alan Turing who hacked Enigma etc. For widespread nouns morphological parsing to merge candidates could be used that are morphological variants.

In conclusion for the 4 general states of Watson, in these states of DeepQA which it draws on systems of the IR-based paradigms are analyzed. Actually, Watson's architectural improvement is its confidence on recommended a large count of candidate answers from text based, then progressive a lot of variety of proof properties to score these candidates again text based. Certainly, the Watson system includes much more components for dealing with complicated and uncommon questions.

Watson and its coginitive capabilities mirrors some of the key cognitive elements of human expertise systems that reason about problems like a human does when we as humans seek to understand something and make a decision we go through four key steps. First we observe visible phonemena and bodies of evidence, second we draw on what we know to interpret what we are seeing to generate hypotheses about what it means, third we evaluate which hypotheses are right or wrong, finally we decide choosing the option that seems best and acting accordingly. Just as humans become experts by going through the process of observation, evaluation and decision making. Cognitive systems like Watson use similar processes to reason about the information in they read. Watson can also do this at massive speed and scale.

2.1.1. How Does Watson Work?

Unlike conventional approaches to computing which can only handle neatly organized structured data such as what is stored in a database. Watson can understand unstructured data which is 80% of data today all of the information that is produced primarily by humans for other humans to consume[7]. This includes everything from literature, articles, and research reports to blogs posts and tweets. While structured data is governed by well defined fields that contain specified information Watson relies on natural language which is governed rules of grammar context and culture. It is implicit, ambiguous, complex and a challenge to process. While all human language difficult to parse. Certain idioms can be particularly challenging. When it comes ro text Watson does not just look for key word matches or synonyms like a search engine. It actually reads it interprets text like a person. It does this by breaking down a sentence grammatically, relationally, and structurally. Discerning meaning from the semantics of the written material. Watson understands context. This is very different simple speech recognition which is how a computer translates human speech into a set of words. Watson tries to understand the real intent of the users language and uses that understanding to possibly extract logical responses and draw inferencest to potential answers through a broad array of linguistic models and algorithms.

2.1.2. How Watson Learns?

When Watson goes to work in a particular field it learns the language, the jargon, and the mode of thought of that domain. Take the term cancer for instance, there are many different types of cancer and each type has different symptoms and treatment. However, those symptoms can also be associated with diseases other than cancer. Treatments can have side effects and affect people differently depending on many factors. Watson evaluate standard of care practices thousands of pages of literature that capture the the best science in the field and from all of that Watson identifies the therapies that offer the best choices for the doctor to consider in their treatment of the patient. With the guidance of human experts Watson collects the knowledge required to have literacy in a particular domain what is called a corpus of knowledge. Collecting a corpus starts with loading the relevant body of literature into Watson. Building the corpus also requires some human intervention to call through the information and discard anything that is out of date poorly regarded or immaterial to the problem domain. Next, the data is pre processed by Watson. Building indices and other metadata that make working with that content more efficient is known as ingestion, at this time Watson may also create a knowledge graph to assist in answering more precise questions.

2.3. How Does The SbQ System Work?

Step 1 is data processing. Data collection (set of documents) is created in IBM Watson. Data that are in the created document set are tagged by their entity type such as person entity type for Ataturk or company entity type for Microsoft. The data model is trained by these tagged entity types.

Step 2 is question classification. Question is parsed to realize which entity type(s) could be contain the possible answer(s). The relevant information about question is proceed by IBM Watson and then we obtain the result data set.

Step 3 is analysis the obtained result data set. Part(s) which contains subject of the question are found in the result data set. Then, developed question answering algorithm that, move on in each place from left of the subject and right of the subject at the same time to find related entity type, is run. All found data related with searched entity type are checked for suitability to define it as possible answer. For instance, if we search “date” entity type since the question is asked the born date of subject with “when”, and also there are 2 same entitiy type in the same part, we define the smallest date as possible answer. Finally, suitable possible answer(s) is/are returned to user.

3. External Interface Requirements

3.1. User Interfaces

The user interface will be worked on web-based systems. In addition to this, there is virtual keyboard. After asking question, the system processes states in section 2, then lists the possible answers to user.

3.2. Hardware interfaces

Enough free space for database size.

3.3. Software interfaces

The project needs to run a web space to access any users,and it needs any web browser such as Google Chrome,Internet Explorer,Mozilla Firefox etc. Moreover, the project needs to analysis some sentences so the system will take to pieces user's question. Furthermore, the system need a storage area to take and use some information for better performance while returning the answer

3.4.Communications interfaces

Admin comminicate with system through web browser.

4. Functional Requirements

4.1.Assistant Tools Use Case for Searching

Use Cases

- Virtual Keyboard.
- Question Based Search.
- Answer Sharing.
- Copy Answer.

Diagram

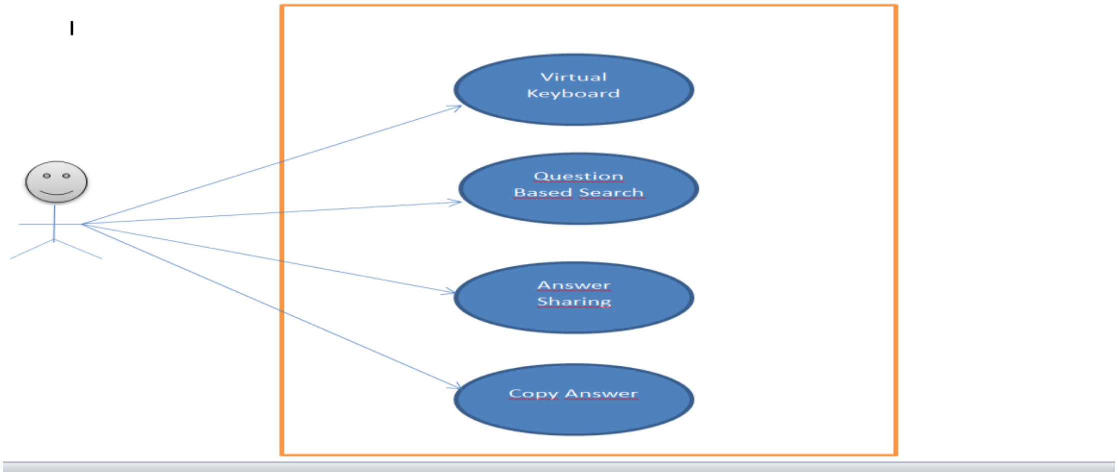

Figure-7: Assistant Tools Use Cases for Searching and happy user 😊

Brief Description:

In Assistant Tools Use Case Diagram(*Figure 1*) explains the basic operations for users. Users are able to use the following functions:

- Virtual Keyboard
- Question Based Search
- Answer Sharing
- Copy Answer

Initial Step by Step Description:

1. User can use virtual keyboard if user doesn't have physical keyboard or users are using mobile phone.
2. If user wants to use Search by Question system, user should ask 5w1h question to system, otherwise system will not return logical answers.
3. User can copy answer of questions with copy button.

5. Performance Requirement

System requirements for any system working with IBM Watson:

1. Operating system: Mac OS X 10.1 and above,Windows 10(64-bit) Enterprise,Education or Pro editions.
2. 4 CPU cores.
3. 8 GB of memory.

6. Software System attributes

6.1.Performance

- We should train to increase performance to find true keys onto asked question on IBM Watson Tool.
- As Watson is trained, accuracy and performance of responses will increase on the system.

6.2.Usability

- If user use 5W 1H(how,who,what,where,when,why) question type, user will obtain a answering centences.
- The question could consist of maximum 160 characters.
- User can ask the question only in English.
- More than one questions could not be asked at the same time.

6.3.Adaptability

- Questions and answers should accommodate to desired format like 5W1H format in English.

6.4.Scalability

- There is no scalability for appealing to the general public.

7. Working Requirement

The entered text should be 5w1h question template. If entered text is not question template, the system can run but it does not return expected result.

8. Development Tools

- Bluemix - IBM Cloud System for use Watson.[8]
- Natural Language Understanding.
- Watson Knowledge Studio - This tool will use train to Natural Language Understanding tool.
- Watson Explorer - This tool will use analytics for unstructured data.
- Watson Discovery – This tool will use to passage retrieval which is mentioned in section 2.1.2
- Used programming languages: PHP, Nodejs

9.SEARCH BY QUESTION INTERFACE

We added the Functional Requirements which is determined SRS(Software Requirements Specification) to design Search By Question Demo. We are designed interface to users.Users shall use this interface and users effectively will benefit this system effice.Firstly, We added Virtual Keyboard for users that use effectively this system.User can reach keyboard on the screen.Following Sytem Interface Show the some usefull tools.In addition to, after asking the question, system finds appropriate results as rules of 5W 1H.

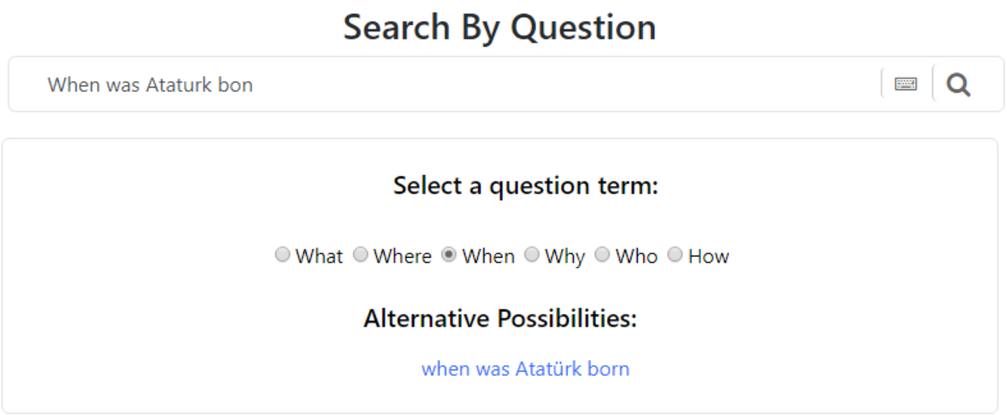

Figure-8: Search By Question Interface to search answers

After user finding the answer ,if he/she needs to see the source,user can click “Go to source of answer” and user can see source of answer. Following figure show the question’s answer and other click button which is “Find all releated sources” that if user need to see another source,user can click this link and user can see all another related sources.

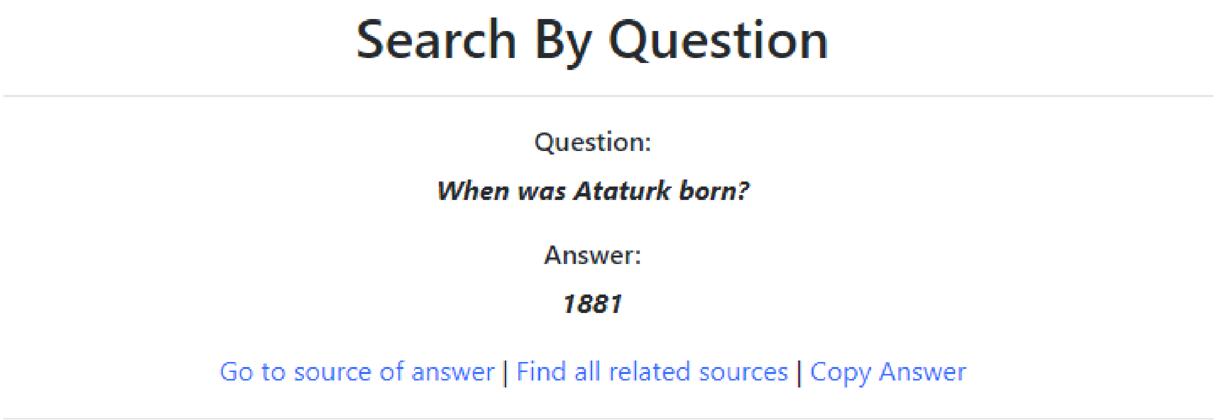

Figure-9: Result Page

Following figure shows the related documents page.

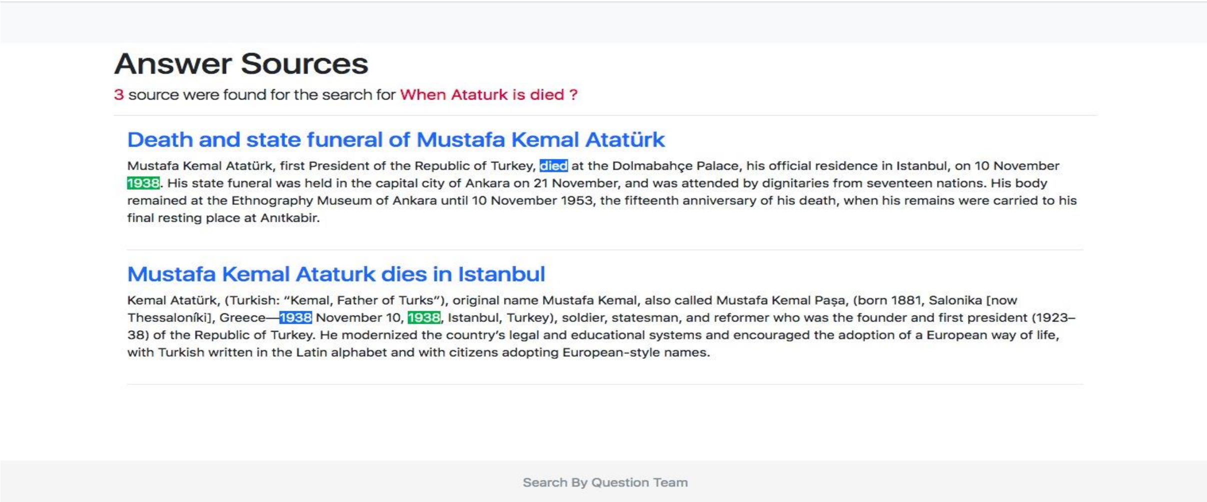

Figure-10: Show Releated Documents

Above Figures explain Search By Quesion's System interfaces and some tools for using system.

10.ENVIRONMENT

This state is about of Search By Question System's environment. IBM Watson, Knowlage Studio and Discovery are important development area for the system.

10.1 IBM WATSON AND KNOWLEDGE STUDIO FOR SEARCH BY QUESTION

Subject matter experts can work together at a a distance to annote with machine learning or rules based methods, without having to create on line of code. Then apply your new model to solutions built with Watson Discovery Services.

Watson Knowledge Studio lets Watson discover meaning full insights in unsuctrucred text without any code.

For example, you need to teach the system what is the virus, crash or something else,for our domain and what is the relationship between these entitites.Traditional technology rolayed only on programming or creating rules to costumize natural language. Now with IBM Watson knowledge studio, you can teach Watson about your specific interest area.

Experts in any field can train Watson to acclerate train you can free any documents used in Watson natural language understanding.Then you can find the results to meet your requirements.Knowledge studio creates machine learning model based on your training Watson can learn its own and identify and relationships go beyeond in these your training.

You use this model to build content analysis application use in Watson discovery or Watson Explorer. Knowledge studio is used to train Watson identify cririrical to stop cyber security attacks.

11. Testing and Release

Test Requirement 5.1.0: System that checks if a connection is established with cloud services.

Test is successfully done. System could provide the connection between application and IBM Bluemix Cloud Service to use IBM Watson tool properly.

Test Requirement 5.1.1: User could not be enter a question if he/she does not choose the question term.

Test is successfully done. Placeholder and virtual keyboard are disabled till choosing any question term from radio buttons.

Test Requirement 5.1.2: When any question term is chosen, warning tooltip have to be invisible.

Test is successfully done. A tooltip appears when user intents to enter a question before choosing a question term. Warning is: “Please select a question term!”. After choosing any question term, tooltip is invisible.

Test Requirement 5.1.3: If entered question is not 5w1h format, application gives an error.

Test is successfully done. After choosing the question term, placeholder is enabled with chosen question term to write a question. If user delete the question term before click the search button, application gives an error that “Your question is not 5w1h format!”.

Test Requirement 5.1.4: Click “Search” button. After clicking, question will sent to the SbQ system backend services and application redirects results page.

Test is successfully done. When search button is clicked, we could achieve the question to make process at the backend and system redirects results page.

Test Requirement 5.1.5: Select “VK” button. After selecting, application will show virtual keyboard.

Test is successfully done. Virtual keyboard is visible when “VK” button is clicked. It could be also used successfully and when click the button again it is invisible again.

Test Requirement 5.1.6: Length of asked question have to be maximum 160 characters.

Test is successfully done. When more than 160 characters are tried to enter in placeholder, system blocks entering any character.

Test Requirement 5.1.7: Select “Go to Source of Document” button from result page. After selecting, application redirects users to datasets page and list document(s) for each result(s).

Test is successfully done. Document that is contain possible answer is seen when “Go to Source of Document” link is clicked.

Test Requirement 5.1.8: Select “Find All Related Documents” button from result page. After selecting, application redirects users to datasets page and list all related results.

Test is successfully done. Documents that is contain related information about question topic are seen when “Go to Source of Document” link is clicked.

Test Requirement 5.1.9: Select “Copy the answer” link from result page. After selecting, application will copy answer to clipboard.

Test is successfully done. When the link is clicked, answer is copied.

Test Requirement 5.1.10: Spellchecker plugin will work when users start to enter text to placeholder. It will check typing errors.

Test is successfully done. It offers alternative possibilities to user word by word.

Test Requirement 5.1.11: Entered question have to be parsing.

Test is successfully done. Question type should be specifyed to realize which entity type(s) could be contain the possible answer(s). When we implement the developed parsing algorithm and print the result on the console log, we see that returned result is expected result of this test.

Test Requirement 5.1.12: Part(s) that contains subject of question must be found in the result data set.

Test is successfully done. The relevant information about question is proceed by IBM Watson, then we obtain the result data set. All Parts which are contain question subject in result data set have to be found to close the find possible answer. When we test the algorithm, all these parts are found.

Test Requirement 5.1.13: Question Answering Algorithm have to be run without any missing.

Test is successfully done. After specifying place(s) that the subject located in , we move on in each place from left of the subject and right of the subject to find related entity type at the same time.

Test Requirement 5.1.14: Found suitable possible answer(s) could be returned to the result page with asked question.

Test is successfully done. Found result(s) and asked question are shown at the result page after all process are done.

Test Requirement 5.1.15: Compatibility for each device have to be provided.

Test is successfully done. Application design is responsive and has same appear in each device.

12. References

[1] Na, Kang, Lee, Lee, S, I, S, J, "Question Answering Approach Using a WordNet-based Answer Type Taxonomy", Semantic Scholar

pdfs.semanticscholar.org/8717/bf98ce28b9a4a33704bd32afb1a5fa3385ab.pdf

Last access date: 19.12.2017

[2] Levow, G, "Question-Answering: Evaluation, Systems, Resources", University of Washington Natural Language Processing Systems and Applications Course Page

faculty.washington.edu/levow/courses/ling573_SPR2011/slides/ling573_class3_qa_systems.pdf

Last access date: 21.12.2017

[3] Dominguez-Sal, Surdeanu, D, M, "A Machine Learning Approach for Factoid Question Answering", The Spanish Society for Natural Language Processing

www.sepln.org/revistaSEPLN/revista/37/17.pdf

Last access date: 21.12.2017

[4] Semantic Scholar , "Learning Question Classifiers: The Role of Semantic Information"

pdfs.semanticscholar.org/f2b1/8264de28827a061fe9e22c437d1f616fdb4a.pdf

Last access date: 22.12.2017

[5] Lin, J, "An Exploration of the Principles Underlying Redundancy-Based Factoid Question Answering", University of Waterloo Computer Science Department

cs.uwaterloo.ca/~jimmylin/publications/Lin_TOIS2007.pdf

Last access date: 22.12.2017

[6] Ta-Cheng Chen, C, "IAs Based Approach for Reliability Redundancy Allocation Problems", science direct

http://www.sciencedirect.com/science/article/pii/S0096300306005042

Last access date: 08.12.2017

[1] Marr, B. "The Rise Of Thinking Machines: How IBM's Watson Takes On The World". Forbes

goo.gl/173y78

Last access date: 27.12.2017

[8] IBM, "IBM Cloud".

https://console.bluemix.net/catalog/

Last access date: 07.12.2017