5-2016

A comparative approach to Question Answering Systems

Josue Balandrano Coronel
The University of Texas Rio Grande Valley

A COMPARATIVE APPROACH TO QUESTION ANSWERING SYSTEMS

A Thesis

by

JOSUE BALANDRANO CORONEL

Submitted to the Graduate College of
The University of Texas Rio Grande Valley
In partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2016

Major Subject: Computer Science

A COMPARATIVE APPROACH TO

QUESTION ANSWERING

SYSTEMS

A Thesis
by
JOSUE BALANDRANO CORONEL

COMMITTEE MEMBERS

Dr. Laura Grabowski
Chair of Committee

Dr. Emmett Tomai
Committee Member

Dr. Zhixiang Chen
Committee Member

May 2016

ABSTRACT

Balandrano Coronel, Josue, A Comparative Approach to Question Answering Systems. Master of Science (MS), May, 2016, 50 pp., 4 tables, 6 illustrations.

In this paper I will analyze the efficiency, strengths, and weaknesses of multiple QA approaches by explaining them in detail. The overarching aim of this thesis is to explore ideas that can be used to create a truly open context QA-System.

The various algorithms and approaches presented in this work will be focused on complex questions. Complex questions are usually verbose and the context of the question is equally important to answer the query as is the question itself. The analysis of complex questions differ between contexts. The analysis of the answer also differs according to the corpus used. Corpus is a set of documents, belonging to a specific context, where we can find the answer to a specified question. I will start by explaining various algorithms and approaches. I will then analyze its different parts. Finally, I will present some ideas on how to implement QA-Systems.

DEDICATION

I would like to dedicate this thesis to Dr. Laura Grabowski, whose support and guidance were imperative to finish this thesis. My entire family who's always been there for me as well as my friends who supported me and let me bounce ideas off of them. I would also like to dedicate this thesis to my future wife Gigimaria Flores Pedraza for her support throughout this endeavor. Finally, I would like to dedicate this to you, the reader, hoping that this work will trigger an idea or an interest on the various topics briefly touched on these pages.

ACKNOWLEDGEMENTS

I want to thank Dr. Laura Grabowski who was able to help me organize my ideas and for her spot on advice. I would also like to thank my committee Dr. Chen and Dr. Tomai to take time and give me advice about this work. Finally , I would like to thank all the great researchers and developers referenced in this work for creating the knowledge that helped me finish this paper.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION AND BACKGROUND

Background

Since the early 1950s there has been research in Natural Language Processing (NLP). NLP refers to the field in computer science which approaches the interaction between computer and human. The main problems of NLP are to extract meaning from natural language, understanding natural language and constructing natural language sentences or passages[1]. This research had a tremendous boost in the mid 1970s when the U.S. Government invested a lot of resources in NLP research (Preeti et al. 2013). Starting in 1992 the Text Retrieval Conference (TREC) (Harman, 2001) co-sponsored by the National Institute of Standards and Technology (NIST) has been a great platform for some of the best improvements in NLP and general information retrieval algorithms and approaches. There has been a tremendous focus on Question Answering Systems (QA-Systems) in the industry as well. Most of the "industry giants" (Google, Amazon, Microsoft, IBM) have realized the power behind a robust system to answer any type of question, be it a simple inquiry or a more complex one. One of the best examples of fully implemented QA-Systems is Watson by the IBM group. IBM has offered the technology behind Watson as a platform for anyone that would like to use it to apply it to different problems. Watson was first developed to compete with a Human Jeopardy champion. In the last few years the IBM team have been trying to implement Watson to help doctors diagnose patients. Diagnosing patients is a very important problem which has been studied for several years now.

Currently there are a few implementations of QA-Systems directed to helping doctors diagnose patients, include askHermes (Cao et al. 2011) and UpToDate (Garrison et al. 2003).

The basic structure of a QA-System consists of three main blocks. The implementation of these blocks can vary tremendously. Take into consideration that classifying the architecture of a QA-System in three main blocks is an oversimplification made to present an introduction to the topic and based on a lot of literature of the subject.

The first block is the NLP block, where the system processes and analyzes the input question, which is in natural language form (Preeti et al. 2013). The main problem here is to convert from syntax to meaning. When working with complex questions there are different approaches for processing the question. Some approaches have explored the follow-up questioning route. Follow-up questions are asked to the user in order to narrow down the answer domain and/or make a better sense of the question asked (McGuinnes, 2004). Other approaches try to classify the question in a subtopic so the answer extraction will be within a smaller corpus (Harabagiu et al. 2006). A corpus is a collection of documents from which the answer to specific question is constructed.

The second block is the document retrieval part of the system. Here the QA-System will retrieve a set of documents where the possible answer might exist. This step has been done in different ways. The simplest approach is to retrieve any document which contains any of the keywords extracted from the input question. This is usually done when working with simple, straightforward, questions. When dealing with complex questions extracting meaningful keywords becomes more difficult since a complex questions is, more often than not, verbose and we could, possibly, extract a large set of keywords from it and not all of these keywords might be meaningful. Another approach is to create regular expressions which will represent the type of answer we are searching for. The type of answer will be determined by the type of question the system gets as input. The more complex and effective methods are the ones that leverage machine learning algorithms. Clustering, Tree Searches and Bayesian Classifications are some of the algorithms used to retrieve a good set of documents for answer extraction.

The last block handles the passage retrieval and answer output. This can be seen as two separate blocks depending on the complexity of the processes. Usually passage retrieval resembles the summarization of documents. There are also different machine learning algorithms used to retrieve the correct passages. Semantic Analysis, Bayesian Classifiers and Support Vector Machines are a few examples of machine learning algorithms use to retrieve passages. I will talk more in depth about these algorithms in chapter II. This is done because when answering complex questions the answer usually does not lie in one single document. The various machine learning algorithms are used to identify the correct context and meaning of the passages. Once the correct passages are retrieved the system constructs the answer.

Motivation

In this paper I will analyze three different approaches to answer extraction in Question Answering Systems (QA-Systems) for complex questions. I will analyze the approaches, their strengths and weaknesses. I will then present different suggestions on how to implement QA-Systems.

The motivation for this paper is to better understand QA-Systems to eventually approach a good implementation for a QA-System. The QA-System subject brings together multiple Artificial Intelligence (AI) topics. The first step of a QA-System is to analyze a natural language input, this is called Natural Language Processing. The second step of a QA-System is to retrieve a set of documents (called *a corpus*) in which the system will search for the answer. There are different approaches for this step that use a variety of machine learning algorithms. Finally the QA-System has to retrieve multiple passages and then construct an adequate answer for the inputted natural language question. This last step uses a mix of NLP, machine learning algorithms and statistical analysis.

As we can see QA-Systems constitute a cornerstone in Artificial Intelligence. This can be seen as one way to communicate with artificial entities. There has been some progress in this topic. One of the best examples of a fully functional QA-System is Watson from IBM (Ferrucci et al. 2010). This system is capable of answering factoid questions in an average of three seconds, as well as other, more complex, questions. Factoid questions are questions which asks for specific information and can be answered with a simple fact, e.g.:

- *When was the declaration of independence announced?*. Answer: **July 4, 1776**.
- *Who was the first president of the United States?*. Answer: **George Washington**.

The IBM team is currently trying to apply Watson to different, more complex, questions and corpus. There are also multiple QA-Systems focused on medicine to help doctors diagnose patients faster (Cao et al. 2011). Currently, the Text Retrieval Conference (TREC) is the main venue where a lot of new development for question answering and information retrieval from text is presented and implemented. A lot of independent companies are investing resources into QA-Systems because of the value that brings to users.

This paper focuses on different approaches to answer Complex Questions. I refer to Complex Questions as questions which are more verbose and contain a more specific context instead of just a short sentence for the question. For example; *I've been running three times a week for the past 4 months, how can improve my endurance?*. As we can see, in this question, we have more information to make a sense of what the inquirer is talking about.As we move to complex questions we move to a more real, and intuitive, communication. The type of questions we, as human beings, use to interact with each other are usually complex questions. The main difficulty when processing this type of questions is to fully understand the context of the possible answer as well as the meaning of the question itself. Even when asking questions to another human being we tend to make follow-up questions to fully understand what the inquirer is referring to.

Central problems and Questions

Question Answering Systems (QA-Systems) involve multiple problems which can be solved in different ways. We can take a look at the two main classifications of QA-Systems: Closed Context (closed domain) and Open Context (open domain). These two classifications refer to the type of knowledge base (corpus) on which a QA-System works. Closed Context is specific to a subject. This means that the corpus and questions used by the system are of a specific domain. For example, there are QA-Systems developed specifically to answer biomedical questions (Athenikos, 2010). There are also QA-Systems developed in the clinical question domain (Terol et al. 2010), (Cao et al. 2011). There has also been a lot of work done in open domain QA-Systems. These systems are designed to answer any type of question given to them. For instance, there are systems focusing on factoid question answering (Tahri et al. 2013, Ferrucci et al. 2010). There is also a lot of research and development around complex questions (described earlier) (Diekema et al, 2004) and temporal questions (Saquete et al, 2009). In a temporal question the answer is a summary of a timeline of a specific or multiple events. These are just some examples of the different subtypes of QA-Systems that can be implemented.

The problem of answering questions is not only about the domain pertaining to questions but also on the form and context of the question. This is the more general issue when diving into

QA-Systems. Since the first step of a QA-System is processing the inputted question it is imperative to tackle this problem in an efficient way. This processing takes place by using Natural Language Processing (NLP). NLP is a very large topic and there are many methods to analyze a question. Some implementations use context free grammars, word relationships (Ferrucci et al. 2001) or custom question models classification (Oh et al. 2003), to mention a few. This classification is also done with the help of machine learning algorithms like Support Vector Machines, Neural Networks or even simple Bayesian Classification. The solution of this problem will lead the system to pre-qualify the type of answer it needs to yield. For example, if the question processing reveals that the inquiry is about a person, then the search algorithm can focus on personal names.

The next step presents a very interesting problem to solve. This is when the system has to retrieve a set of documents in which the answer might be. This is also done with the help of NLP and different machine learning algorithms. By now the QA-System already has more information to rely on. The system already knows which type of sentence it is looking for and some of the keywords that sentence (or paragraph) it should contain. The difficulty of this problem grows depending on the complexity of the answer and, although not necessarily, the complexity of the question itself. e.g. A factoid question may only need an answer of a name, place or date (Tahri et al. 2013) whilst a clinical question would need an answer composed of a definition and possibly a list of symptoms (Garrison et al. 2003).

Finally, the QA-System needs to render a human readable answer. There are different techniques used in this step e.g. Summarization, language models, etc… This problem is very interesting because it has to yield an insight to the entire process.That is, if an answer is not well constructed it might not reflect the information easily.

In this paper I will analyze the following questions:

- What are some of the different methods used to solve the previous problems?.

Multiple and different approaches to answer complex questions have been proposed in the past. These approaches range in complexity as well as in the composing elements. Some approaches use statistical methods, such as Naive-Bayes, or they use more complex methods such as Lexical Semantic Analysis. There have been implementations which are more hybrids and use a combination of different algorithms through each one of the steps in the QA-System. I will introduce and analyze some of these approaches.

- What are the difficulties of implementing different methods to solve the previous problems?.

Information retrieval from natural language can be categorized as an open problem. This is because it is very difficult to construct one answer for every possible case. Extracting information from natural language can get very complicated given the nature of languages. Because of this, there is still a lot of research been done in this topic and improvements are been proposed.

- How can we make a step forward to create a truly open context QA-System?.

In order to create a truly open context QA-System there are a lot of variables one must take into consideration. Not only about the data that's been analyzed but also about the question that needs an answer as well as the user who is requesting the information. I will talk about different variables that, in my opinion, must be taken into consideration in order to keep moving forward and create a truly open context QA-System

CHAPTER II

METHODS

Automatic Text Summarization

In this modern age we have easy access to an amazing amount of data. Search engines like Google or Bing allow us to search through most of the data available out there. Nonetheless, search engines do not analyze the entirety of the information retrieved. Search engines use a keyword approach when searching for relevant documents. There is still a good amount of processing of the initial query as well as the user's data in order to search and score relevant documents. This represents a problem because the user is left with only a list of probable relevant documents, or documents where the actual answer to the initial query may reside. The user still needs to go through this documents and analyze them in order to extract the desired information. It is evident that this is an overload of information for the user. A more automatic approach is needed. This is the main focus of Question Answering Systems (QA-Systems)

A QA-System analyzes a query and then tries to extract an answer using this analysis and one or multiple documents where the answer may exist. We can refer to this information extraction as a summarization of one or multiple documents. This is categorized in two classifications :

- Single-Document summarization
- Multi-Document summarization

(Hacioglu et al. 2004)

In single document summarization an answer is extracted using only one document. This approach can be used when trying to answer simple questions, e.g. *What's the capital of Canada?*. In multi-document summarization a set of multiple documents are retrieved and analyzed in order to extract an answer. The set of documents is referred to as *corpus.* Multi-document summarization is used when trying to answer more complex questions, e.g. *What were the main causes for the great depression?*. As we can see in this query a QA-System would need to extract the information from multiple documents. Multi-document summarization also poses another problem, how to present the extracted answer. There are different ways to construct a summary:

- Generic summary
- Query based summary

(Hacioglu et al. 2004)

A generic summary is the general idea of a specific document. Understanding the topic of the document is imperative in this type of summarization. This is the type of summary a human being does when reading a document. There has been different approaches to create a generic summary. Carbonell and Goldstein (1998) created one of the best approaches using Maximal Marginal Relevance (MMR) which uses the vector-space model of text retrieval. Although this approach is very useful to retrieve the general idea of a document QA-Systems have been inclining towards query based summarization in order to give a better answer to a question. In a query based summarization the system focuses on searching information specific to the query, rather than the document itself, to return the desired answer. There are also different approaches to present the summary:

- Abstract Summary

- Extract Summary

(Hacioglu et al. 2004, Zechner et al. 1998)

An abstract summary can be constructed of words and sentences that do not appear in the corpus while an extract summary focuses on weighting words, sentences and/or paragraphs by their relevance in the central topic and/or query and creating a summary using the previously weighted words, sentences and/or paragraphs. Extract summarization is simpler than abstract summarization, given the fact that in order to create an abstract summary it is necessary to understand the corpus, word context relevance and grammatical structure. The aim of QA-Systems is to use a query based summary using a multi-document corpus and presenting the answer in an abstract summary.

Knowledge Based

In a knowledge based system, a database with possible answers is constructed before any queries are made and the answer(s) returned only exists in this database. Usually, this approach is very fast and is one of the first methods used in Question Answering Systems (QA-Systems) (Terol et al. 2007). The idea behind this method is to feed a set of documents into the system and analyze them. Within this analysis multiple techniques may be used to classify the important information and construct possible answers. The methods to analyze the knowledge base include natural language processing (NLP) (Terol et al. 2007, Tahri et al. 2013, to mention a few) and sentence extraction and/or sentence construction (Carbonell and Goldstein 1998, Kim et al 2001, to mention a few). Analyzing a knowledge database beforehand has shown to be a very fast approach to question answering. This allows to draw different important features from the possible answers as well as help to classify the query given. In this type of system most of the processing takes place when analyzing the given question and use the previously extracted information to return an answer (Terol et al. 2007, Liu et al. 2015).

In past years multiple approaches using knowledge bases to answer questions has been implemented (Liu et al. 2015, Sheldon 2011,).

We can classify crowd question answering platforms as Knowledge Based like Yahoo! Answers[2] or Stack Overflow[3]. In this platforms no automatic implementation is done, rather users pose questions which are then answer by other users. These platforms do not implement any of the methods listed in this paper, it is still worth mentioning because they serve as a service for users to find answers to their questions as well as provide a database that could be further analyze to construct QA-System using the different approaches mentioned in this paper (Liu et al. 2014).

Another type of knowledge based system focuses on automatic answer retrieval. The system has access to a database of questions and answers which are used to classify the question given to the system and the corresponding answer. An example of knowledge base system is Watson IBM (Ferrucci et al, 2011). Watson had a corpus constructed from encyclopedias, news articles, thesaurus and some literary works. Another example is MAYA (Kim et al, 2001). This system constructed a database of questions and answers as its first step. The answers are handled as passages. When a question is inputted into the system, it will search for the passage that relates the most to the question and return it.

As we can see there are multiple advantages to knowledge based systems. When a question is posed the system only has to compare the possible answers in its database to the inputted query and return that answer. Using a previously processed knowledge base presents an improvement in answer extraction efficiency. Of course, extracting the answer from the knowledge base taking into consideration the question (Carbonell et al 1998, Kim et al. 2001). is just the last step of the process. Before the system can have all the necessary information a knowledge base system can grow very complicated depending on the methods used to extract possible answers from the corpus. In addition, the query given to the system can be analyzed to extract the relevant keywords as well as the relationship between the extracted keywords and the answer. This can result in a highly efficient and accurate QA-System. IBM Watson was able to answer Jeopardy questions in approximately 3 seconds and was able to beat the human Jeopardy champion.

It is also important to denote the limits of such systems. First of all this type of system can only use answers in the knowledge base to answer a question. Typically this approach is used to answer questions in a specific closed domain, meaning that it will only answer questions regarding specific topics in its database (Liu et al. 2014, West et al. 2014, Link 2011, to mention a few). This restriction can make this approach particularly limited in the verbosity and the information the answers given can transmit.

Question Decomposition

Question decomposition refers to the technique of separating a complex question in smaller questions (Harabagiu et al. 2006). Usually, these resulting questions are very simple ones, i.e. factoid questions or questions for which answers are simple terms or simple sentences. This approach relies on the successful recognition of relationships between words and concepts present in the initial query and the corpus used. Question decomposition can be accomplished by analyzing important keywords in the question and the corpus as well as more robust relationships like semantic or grammatical weight of the keywords relationships between document or complete sentences. Usually, a statistical approach for answer extraction is used (Yen et al. 2013, Tellex et al. 2003, Harabagiu et al. 2006, to mention a few).

The work of Harabagiu et al (2006) is a great example of question decomposition. Their approach consists of three facets:

- Question decomposition
- Factoid Answering
- Multi-document Summarization

Their approach starts by identifying the topic and the relevant documents, then a graph consisting of the relations between key concepts pertaining to the topic of the question and sub questions is created. Once this graph is obtained, complex questions are decomposed in smaller simpler questions using a Markov chain following a random walk. Finally text summarization techniques are used to retrieve relevant passages and to construct an answer.

Harabagiu et al. (2006) approach leverages on the ability to easily detect the expected answer type of a factoid question. To detect the expected answer type a semantic class is assigned to each factoid question. The approach consists of four steps.

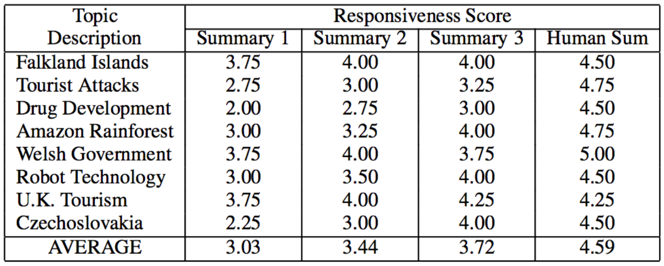
First, the given question is processed to derive the corresponding relations. The question is analyzes lexically, semantically and syntactically in order to identify the relationships between keywords. A Brill tagger (Brill, 1995) is used to analyze the question lexically. Then a probabilistic parser (Collins, 1999) is used to yield the syntactic parts between the question and their relations, e.g. nouns, verbs, etc. After this analysis WordNet is used to determine if any nouns are a nomalization form of a verb, this parts are regarded as redundant and not taken into consideration. Using the WordNet database the syntactic relation between verbs and nouns are drawn.

Second, for each relation constructed in the first step a question is created that involves this relation. The question is created by constructing a sentence which could be the answer to the previously created question using an approach proposed by Harabagiu et al. 2005. This approach of sentence construction is done by further generalizing the relations previously drawn and identifying important entities (e.g. proper names, dates, verbs, etc.).

Third, a new set of relations is created by combining the relations from the previous two steps. Meaning that this new set of relations will contain relations between the lexical and semantical analysis of the given question as well as the relations with the possible deconstructed factoid questions.

Finally, a formalization of the process is proposed by "doing a random walk on a bipartite graph of questions and relations" (Harabagiu et al. 2006). The random walk is stopped after *k* number of iterations.

The evaluation of the resulting decomposed questions was evaluated against decomposed questions created by humans and against questions created from the abstracts of the documents used to answer the given question. Harabagiu et al. (2006) approach was submitted to the DUC-2005 question-directed summarization task and evaluated by NIST editors. This next table shows the results.



Topic Description	Responsiveness Score			
	Summary 1	Summary 2	Summary 3	Human Sum
Falkland Islands	3.75	4.00	4.00	4.50
Tourist Attacks	2.75	3.00	3.25	4.75
Drug Development	2.00	2.75	3.00	4.50
Amazon Rainforest	3.00	3.25	4.00	4.75
Welsh Government	3.75	4.00	3.75	5.00
Robot Technology	3.00	3.50	4.00	4.50
U.K. Tourism	3.75	4.00	4.25	4.25
Czechoslovakia	2.25	3.00	4.00	4.50
AVERAGE	3.03	3.44	3.72	4.59

Table 1: Responsive score for summaries and human summaries. (Harabaiu et al. 2006)

A score from 1 to 5 is given to each of the summaries depending on the amount of information given by the summary. A score of 1 represent the least informative summary whilst a score of 5 represents a very informative summary.

As we can see this approach is highly efficient to decompose question and draw informative relations between the decomposed question and the given question. The efficiency of this method relies on the efficiency of the technique used to decompose questions as well as the processing of the documents in the corpus (Harabagiu et al. 2006).

Graph Based

Graphs allow us to construct relationships between entities taking into consideration multiple elements as well as comparing and searching very efficiently. LexRank (Erkan et al, 2004) and TextRank (Otterbacher et al, 2005) proposed graph based approaches to multi document summarization. These approaches have been very successful.

Erkan and Radev (2004) use a graph based approach to rank sentences to produce the desired summarization. The graph constructed by their implementation contains similarity values between sentences in the set of documents. In these graphs each node is a sentence and each edge is the cosine similarity between two sentences represented as nodes in the graph.

Otterbacher et al (2005) presented an extended LexRank implementation. The improvement in this extended LexRank implementation is the use of a random walk model using the similarities between a given pair of sentences as well as similarities between sentences and the main topic description and/or question. In the extended LexRank approach the model used goes one step further to select relevant sentences. The idea is that if a sentence scored a high relevance to the input question, by using similar keywords, then a related sentence should also be relevant and not necessarily share similarities with the question's keywords. The system calculates the relevance of a sentence to a question using this equation:

$$\mathrm{rel}(s|q) = \sum_{w \in q} \log(tf_{w,s} + 1) \times \log(tf_{w,q} + 1) \times \mathrm{idf}_w,$$

Where $\boldsymbol{tf}_{w,s}$ and $\boldsymbol{tf}_{w,q}$ represent the number of times a word (*w*) appears in the question (*q*) and the sentence (*s*) that are being analyzed. This is based on (Jones, 1972) Allen et al (2003) and has been proven successful in query-based sentence retrieval. With this calculation the system then computes the score of a sentence (*s*) given a question (*q*) as the summatory of its relevance to the question and the similarity to the rest of the sentences in the cluster (*C*):

$$p(s|q) = d \frac{\mathrm{rel}(s|q)}{\sum_{z \in C} \mathrm{rel}(z|q)} + (1 - d) \sum_{v \in C} \frac{\mathrm{sim}(s, v)}{\sum_{z \in C} \mathrm{sim}(z, v)} p(v|q),$$

Where *d* is what Otterbacher et al refer to as "question bias" and it is determined empirically. The equation uses denominators to normalize the values. The similarity between two sentences is calculated by using the cosine measure weighted by Inverse Document Frequency (IDFs):

$$\mathrm{sim}(x, y) = \frac{\sum_{w \in x,y} tf_{w,x} tf_{w,y} (\mathrm{idf}_w)^2}{\sqrt{\sum_{x_i \in x} (tf_{x_i,x} \mathrm{idf}_{x_i})^2} \times \sqrt{\sum_{y_i \in y} (tf_{y_i,y} \mathrm{idf}_{y_i})^2}}.$$

The idea of using IDFs to measure relevance between two sentences relies on diminishing the weight of terms that occur more frequently and maximized the weight of terms that occur rarely. The use of IDFs was first proposed by Sparck Jones, K. (1972) and it's been widely used in term weighting.

This approach uses semi-supervised passage retrieval. This is because this method doesn't need a large set of training data and it only has one tune parameter. This system, also, doesn’t rely on the structure of the language and doesn’t leverage on linguistic resources, e.g. Natural Language Processing. This also means that grammatical understanding of the information is missing from the information retrieval and the distinction of a sentence meaning can be lost. None the less, this method has been proven successful on complex question answering and because of its lack of reliance on language grammar and structure it could be implemented easily in broader domains.

Latent Semantic Analysis

Latent Semantic Analysis (LSA) is a technique used in Natural Language Processing (NLP) which analyses relations between a set of documents and terms within these documents. This is done using a statistical approach by producing a set of concepts pertaining to the terms and documents in question. LSA is based on the assumption that words with similar meaning or with a meaningful relation will appear in similar pieces of text (Deerwester et al. 1990).

When answering complex questions it is necessary to analyze complete documents. This can be very resource consuming and complex. Segmenting documents into relevant passages has been proven to be very efficient in information retrieval (Deerwester et al. 1990, Hofmann et al. 2001). With this approach, entire documents are separated into smaller passages which makes it easier to handle and analyze. These passages can be used to rank the document’s relevance to a topic. Also, by semantically analyzing and ranking passages we can extract the most relevant information in a given document. This gives the ability to easily construct answers from passages. Constructing answers from passages has been proven to be more efficient when answering complex questions (Oh et al. 2007). There are three main approaches for this technique.

- Using structural information of a document for passages (Oh et al. 2007).

- Defining passages of a fixed length (Chakik et al. 2004).
- Leveraging in topical cues or semantics to identify meaningful passages (Brants et al. 2002).

One of the earliest approaches was done by Hearst et al. (1994). A document is broken into smaller segments of size *N*, ranging between three to five sentences. The resulting segments are then represented as vectors. Cosine similarity is used to identify potential topic boundaries with the help of a similarity curve.

An improvement in LSA approach has been proposed. Probabilistic Latent Semantic Analysis (PLSA) (Hofmann et al, 2001). Here the documents are split into smaller passages and represented as co-occurrence keyword vectors. These vectors are then expanded using a mixture model in order to extract information on semantically similar words.

Oh et al (2007) use documents from Pascal Encyclopedia as the corpus. They realized there are different features in this type of document. The articles usually contain a topic, summary and content. The sentences in each article usually grow in complexity depending on the topic being explained. That is, simpler topics contain smaller, less complex sentences than more complicated topics. The researchers also observed that paragraphs in the corpus may contain multiple topics. Based on these features, Oh et al (2007) constructed their approach. Topics are automatically retrieved from each sentence and then used for creation of passages. These passages are then used as meaningful information extracted from the documents and for answer construction to further improve accuracy in a QA-System.

Assuming that meaningful topics for a specific domain are frequently used through the body of a document or documents, Oh et al. (2007) proposed a method that selects sentence topics by looking into existent keywords in a document which might be meaningful. These keywords are selected leveraging in a lexical database (Korean Lexical Concept Net for Nouns or LCNN). Candidate sentences for a sentence topic are selected by analyzing the term frequency in that specific domain. With this information, a hierarchy of sentence topics was constructed and manually tuned to remove unnecessary or redundant topics.

The process proposed by Oh et al. (2007) is constructed in two steps:

- Topic Assignment
- Sentence Organization

In the first step, sentences are classified into sentence topics and in the second step sentences are grouped into semantic passages. Because the sentences contain a small number of terms, extra features are necessary for the sentence topic classification. This led into using a Maximum Entropy approach (Oh et al. 2007).

Maximum Entropy (ME) is used when estimating probability distributions or modeling random data. The main concept in ME is that the distribution can be measured by entropy. In order to successfully use ME the data needs to be correctly labeled and expectations of distribution for features needs to be calculated. ME is specifically useful when analyzing data with multiple features.

In order to apply ME to sentence classification a real-valued function is used. The following model is used to implement ME:

$$p^* = \frac{1}{Z(s)} \exp\left(\sum_i \lambda_i f_i(s,t)\right)$$

Where $p^* = p(t \mid s)$ this is the probability that a sentence is classified under a topic *t* given a context *s*. $\boldsymbol{\lambda}$ is an adjustable value pertaining to the *n* feature function. The *f(s,t)* function is the feature function used where:

$$f(s,t) = \left\{ \begin{array}{ll} 1 & \text{if } t = z \text{ and context(s)} \supset \{x, y\} \\ 0 & \text{otherwise} \end{array} \right\}$$

And *Z(s)* is simply a normalizing value to ensure proper probability:

$$Z(x) = \sum_{c} \exp\left(\sum_{i} \lambda_i f_i(x, y)\right)$$

Using ME focuses on sentence classification and topic to term distribution. Sentence classification poses an interesting problem because of the usual short length of sentences. Because of the short amount of terms in sentences more features are taken into consideration instead of just using specific terms. Sentence patterns and extended verbs are also taken into consideration when analyzing each sentence. The reason why these features are chosen is because there are specific sentence patterns in encyclopedia articles. In these patterns the verb plays an important role. In order to be able to use a relative small amount of training data, extended verbs are used to handle possible verbs found in the actual data (Oh et al. 2007).

In order to generate semantic passages, each one of the sentences in a document is deconstructed into smaller, simpler, sentences. This process involves linguistic analysis, POS tagging, word sense disambiguation (WSD) and AT tagging. The Korean LCNN and the location of a word is used for disambiguation. The decision to use these specific approaches for linguistic analysis is entirely empirical. After this analysis the sentences are passed into the sentence classifier and then the classified sentences are clumped together to create semantic passages (Oh et al. 2007).

As we can see LSA (Oh et al. 2007) and PLSA (Hofmann et al. 2001) can be used for QA-Systems in different ways. It is important to note that the semantic meaning of the information is very important when answering complex, or any kind, of questions. Not only this, but also LSA has been used in other type of applications where information retrieval is paramount. By analyzing the information in this manner the efficiency of a QA-System can be greatly improved (Oh et al. 2007).

It is also important to note some disadvantages of this method. The main one is the amount of human intervention involved. A large amount of data needs to be correctly labeled and calibrated in order to successfully train the system. Take into consideration that not only this approach suffers from this disadvantages. Any other type of trained algorithm will be prone to this.

Another important point to raise is the specificity of the data that can be used. Oh et al (2007) used articles from an encyclopedia which is data structured in a specific way to some extent. Although this approach can be very accurate when the structure of the data could be known or inferred as well as the possibility to label and calibrate the training algorithm, using any other type of random data with this method is not an easy exercise.

Latent Dirichlet Allocation

In information retrieval (IR) has been extensive research to semantically analyze text in order to extract latent (not observed) features, such as PLSA (Hofmann et al. 2001) and LSI (Oh et al. 2007) previously discussed in this paper. LSI and PLSA approaches reduce documents into a latent semantic space. LSI uses singular value decomposition to capture most of the variance in the collection of documents. Deerwester et al. (1998) have proven that LSI can also capture basic linguistic features such as synonymy and polysemy. A significant step further from LSI was proposed by Hofmann (2001) in which every word from a document is modeled as a sample of a mixture model. In these mixture models its components are multinomial random variables which can be represented as topics.

"While Hofmann's work is a useful step toward probabilistic modeling of text, it is incomplete in that it provides no probabilistic model at the level of documents." (Blei et al. 2003). De Finetti (1990) established that a collection of exchangeable random variables can be represented as a mixture distribution. De Finetti's theorem suggest that to consider exchangeable representations for documents and words we need to take into consideration mixture models that reflect this exchangeability of word and documents. Blei et al. (2003) propose Latent Dirichlet Allocation (LDA) based on the previously discussed ideas.

Blei et al. (2003) define the following terms in order to present the proposed LDA algorithm:

- A word is an item of a vocabulary indexed by {1, … V}
- A document is a sequence of N words, $\mathbf{w} = \{w_1, w_2, \ldots , w_M\}$. Where w_n is the n-th word in a sequence.
- A corpus is a collection of M documents denoted by $D = \{w_1, w_2, \ldots , w_M\}$ (Blei et al. 2003)

LDA is a generative probabilistic model of a corpus that assigns high probability to members of the corpus as well as other similar documents. The basic idea is that documents are represented as random mixtures over latent topics. Each topic is a distribution over words (Blei et al. 2001). Also, the word topic is used to identify and label a group of words with similar meaning over the corpus.

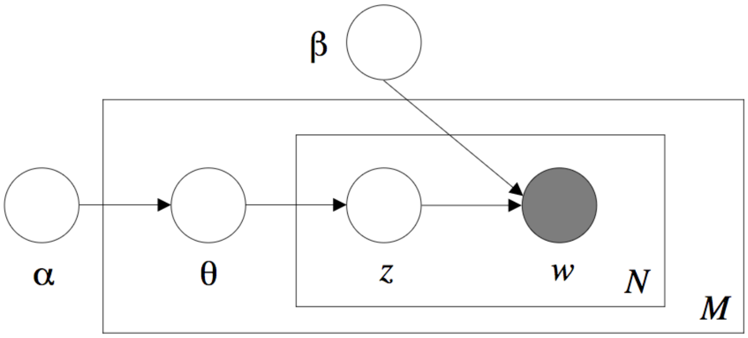


Figure 1: A probabilistic graph representation of LDA (Blei et al. 2001)

In Figure 1 the LDA algorithm is represented in a graphical manner. Each one of the squares represents repetition over documents *M* and words *N*. LDA is a three level model. The parameters **α** and **β** are sampled once on the corpus level. The ***θ*** parameters is sampled once per documents and, *z* and *w* are word level variables and are sampled once for each word in the document (Blei et al. 2001).

LDA assumes the next generative model for each document:

1. Choose N ~ Poisson(ε)
2. Choose ***θ*** ~ Dir(**α**)
3. For each N word in w_n:
    a. Choose a topic z_n ~ Multinomial(***θ***)
    b. Choose a word w_n from $p(w_n|z_n, \boldsymbol{\beta})$

    (Blei et al. 2001)

This could be more easily explained like this:

1. Choose the number of words a document could have, according to a Poisson distribution.
2. Choose a topic mixture for the document based on a Dirichlet distribution (Dir(**α**)). This assigns a possible distribution of each of the topics. It is important to remember that topics in this context is just a label and not necessarily a specific word. The number of topics is empirically set.

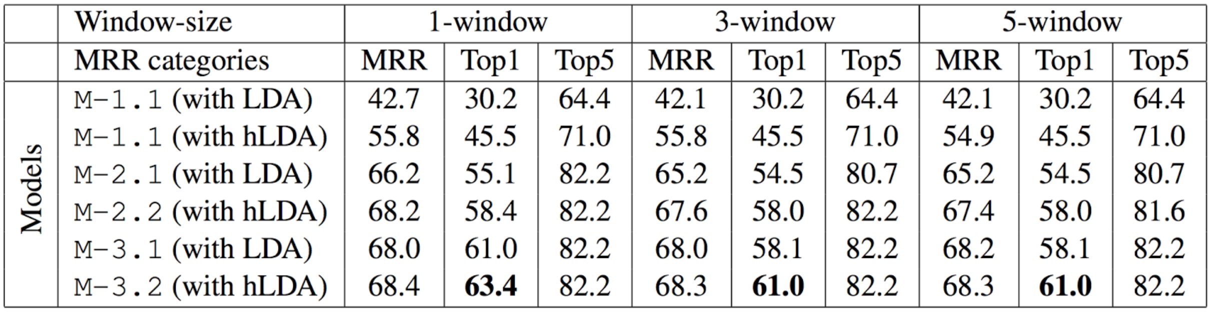
3. Each word *N* in a document w_n is generated:
    a. Choose a topic based on the Multinomial($\boldsymbol{\theta}$) probability calculated before
    b. Choose a word based on the Multinomial($\boldsymbol{\theta}$) probability conditioned by the topic z_n. Meaning, that a topic is selected and the, based on a Multinomial probability, a word, which exists in the vocabulary, is selected.

LDA assumes this generative model for each document and then it tries to infer the topics that exists in the latent space (not observed). We can see now that a topic is a label which will give us a set of words that are correlated (Blei et al. 2003).

LDA has been used in different applications to classify information based on variables that are not observed on the corpus but can be inferred based on the words that appear throughout the corpus. This model has not only been used in text information retrieval (Daniel et al. 2014, Li et al. 2014) but also to classify images (Rasiwasia et al. 2013, Lienou et al. 2010) and other type of information.

A great example of using the LDA model for question answering can be seen in the work by Celikyilmaz et al. (2010). Celikyilmaz et al. (2010) approach constructs an LDA model based on the given question and the set of candidates passages. They build the passages from a corpus retrieved by a keyword query and separating these documents into sentences. This passage generation technique yields approximately 2500 passages for each question. Celikyilmaz et al. (2010) then calculate a similitude metrics to classify the passages that best answer the given question. These similitude metrics are calculated using the information radius. They first calculate the information radius similitude between the topics and then calculate the similitude between passage and topic. Finally, using these two similitudes they rank passages to create an answer to a given question. Celikyilmaz et al. (2010) also use a hierarchical LDA which differs from LDA in that it represents topics as a hierarchical structure. This gives more depth to the meaning of each topic

For experimentation Celikyilmaz et al. (2010) use a data set from TREC 2004 using different passage lenghts, represented as window sizes, and then apply a Mean Reciprocal Rank (MRR) to evaluate the results. The next table presents they results using LDA and hierarchical LDA.



	Window-size	1-window			3-window			5-window		
	MRR categories	MRR	Top1	Top5	MRR	Top1	Top5	MRR	Top1	Top5
Models	M-1.1 (with LDA)	42.7	30.2	64.4	42.1	30.2	64.4	42.1	30.2	64.4
	M-1.1 (with hLDA)	55.8	45.5	71.0	55.8	45.5	71.0	54.9	45.5	71.0
	M-2.1 (with LDA)	66.2	55.1	82.2	65.2	54.5	80.7	65.2	54.5	80.7
	M-2.2 (with hLDA)	68.2	58.4	82.2	67.6	58.0	82.2	67.4	58.0	81.6
	M-3.1 (with LDA)	68.0	61.0	82.2	68.0	58.1	82.2	68.2	58.1	82.2
	M-3.2 (with hLDA)	68.4	**63.4**	82.2	68.3	**61.0**	82.2	68.3	**61.0**	82.2

Table 2: Celikyilmaz et al. (2010) MRR results using a dataset from TREC 2004

As we can see LDA has been successfully used in QA-Systems. This reflects the improvement achievable by using LDA models to infer topics from a corpus. It is important to note some of the disadvantages of this approach. First, we need to take into consideration the complexity of creating a model to correctly infer models from different lexically structured corpuses. The complexity of topic inference in different domains can be seen when choosing how many topics a corpus could have, the correct hierarchy and importance of the inferred topics as well as the feature extraction of the corpus. Also, the similitud metric calculation can pose another problem Celikyilmaz et al. (2010).

Support Vector Machines

Support Vector Machines are trainable algorithms focused on two-class problems. SVMs have been used to categorize data into two categories. A set of training data is given to the algorithm, each data record is flagged as belonging to one of two categories. The SVM then constructs a model which successfully categorizes new data into one of the two categories. SVMs represent the model as a mapping of data in space where a gap exists marking both of the categories used. The thresholds of this gap are delimited by two vectors. Data inputted into the algorithm is then categorized based on which side of the gap they are mapped to. The training data can be given in the form (Hirao et al, 2002):

$$(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_u, y_u), \quad \mathbf{x}_j \in \mathbf{R}^n,\ y_j \in \{+1, -1\}.$$

Where x_j is a feature vector of the *j*-th sample and y_i is the class it belongs to. We can show the gap in the next figure (Hirao et al. 2002):

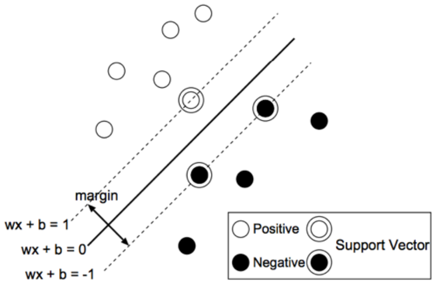


Figure 2: Support Vector Machine (hirao et al. 2002)

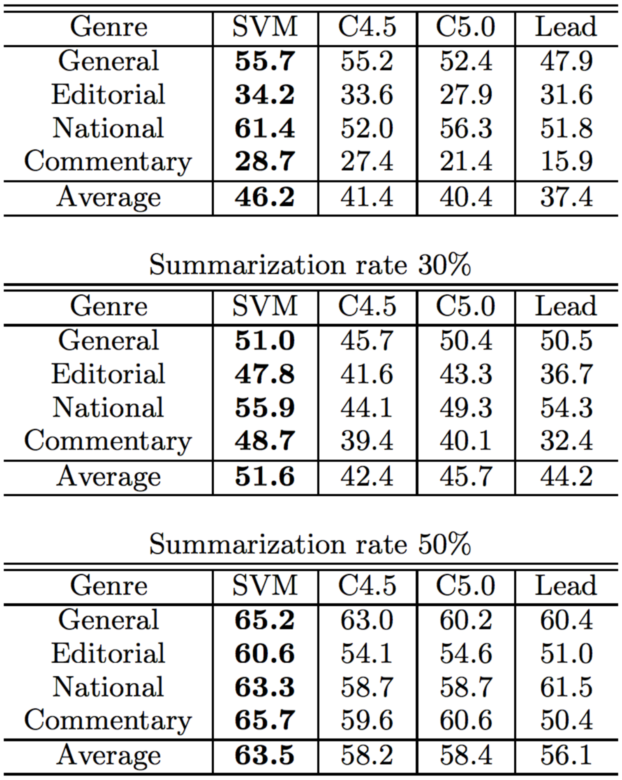
Training data is not always easily classifiable in a linear manner. Because of this a set of slack variable are used to correct the misclassification error. In language processing this vectors are not usually linear. For this a Kernel function can be used. Hirao et al (2007) use a polynomial kernel function for this:

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^d.$$

Support Vector Machines have shown efficient performance when when used for text categorization (Hirao et al, 2002). In this approach multiple features for every sentence is taken into consideration. This features are based off of other past research papers (Zechner et al, 1996.

Nobata et al, 2001. Hirao et al, 2001. Nomoto et al, 1997). Hirao et al (2007) compared decision tree learning, boosting, lead and SVM algorithms to prove the efficiency of their approach. We can see their result in the next table (Hirao et al, 2002)

Summarization rate 10%



Genre	SVM	C4.5	C5.0	Lead
General	**55.7**	55.2	52.4	47.9
Editorial	**34.2**	33.6	27.9	31.6
National	**61.4**	52.0	56.3	51.8
Commentary	**28.7**	27.4	21.4	15.9
Average	**46.2**	41.4	40.4	37.4

Summarization rate 30%

Genre	SVM	C4.5	C5.0	Lead
General	**51.0**	45.7	50.4	50.5
Editorial	**47.8**	41.6	43.3	36.7
National	**55.9**	44.1	49.3	54.3
Commentary	**48.7**	39.4	40.1	32.4
Average	**51.6**	42.4	45.7	44.2

Summarization rate 50%

Genre	SVM	C4.5	C5.0	Lead
General	**65.2**	63.0	60.2	60.4
Editorial	**60.6**	54.1	54.6	51.0
National	**63.3**	58.7	58.7	61.5
Commentary	**65.7**	59.6	60.6	50.4
Average	**63.5**	58.2	58.4	56.1

Table 3: Evaluation Results of Cross Validation (Hirao et al. 2002)

Yen et al. (2013) use SVMs to create a QA-System. Firsts they use SVMs together with word clusters from WordNet to classify the input question. This allows to define what type of answer should the system look for. With this information is easier to extract the necessary features from the given corpus and apply an SVM to classify passages and/or potential answers.

After this question processing, a boolean passage retrieval was implemented (Tellex et al. 2003). The passage retrieval method is used in order to create a smaller corpus where the answer may exists. This way it is faster to search analyze concise information instead of trying to analyze the entire corpus which may contain non relevant data. The boolean model consist in overlapping three sentence and the result is considered a passage. Assuming a document containing *K* sentences

$$\text{Doc} = (S_1, S_2, S_3, \ldots, S_k)$$

The passage is constructed by aggregating three consecutive sentences

$$P_1 = \{S_{1,} S_2, S_3\}, P_2 = \{S_{3,} S_4, S_5\} \ldots P_n = \{S_{k-2,} S_{k-1}, S_k\}$$

Once this is done, a context ranking model (CRM) is applied to the passages in order to rank the passages by their relevance. This is done using short fragments to train the SVM, first. Once the SVM model is created the passages are given to the SVM resulting in a ranked list of passages. This next figure illustrates the architecture of the SVM (Yen et al, 2006)

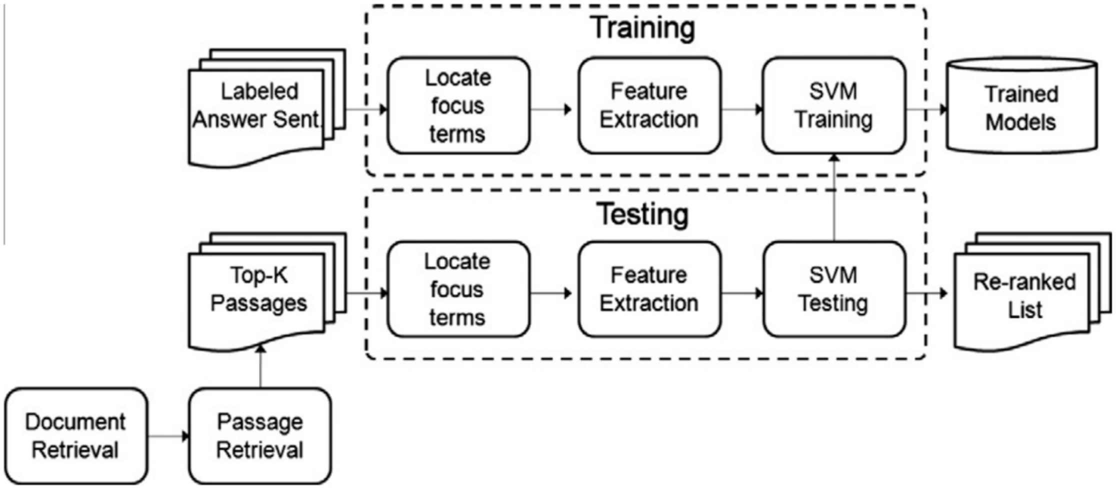
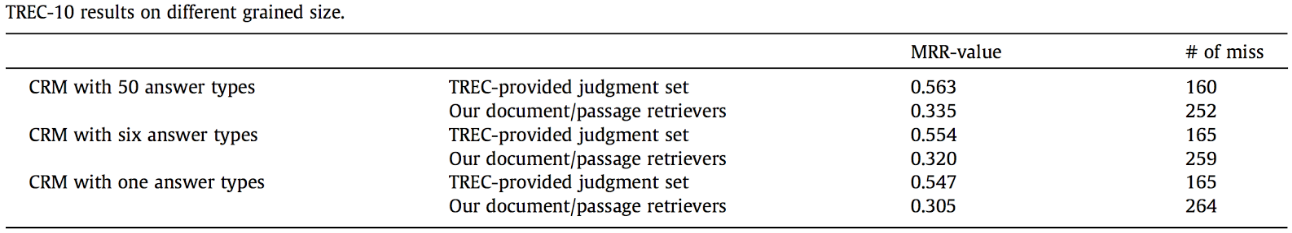


Figure 3: Workflows of the proposed CRM (Yen et al. 2006)

The main reciprocal ranking (MRR) is used in order to evaluate the accuracy of the system. The corpus used is the data from TREC-10 track. The MRR evaluates how accurate the ranking of the actual answer is. This means that if the system returns the answer as the first ranked passage then it is given a 1. If the answer was ranked as the fifth passage then it is given a 1/5th. This can be defined as the following equation (Yen et al. 2006).

$$\text{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{ranki},$$

The overall performance of this approach, taking into consideration the data from the TREC-10 track, is illustrated in the next table (Yen et al, 2006)



TREC-10 results on different grained size.

		MRR-value	# of miss
CRM with 50 answer types	TREC-provided judgment set	0.563	160
	Our document/passage retrievers	0.335	252
CRM with six answer types	TREC-provided judgment set	0.554	165
	Our document/passage retrievers	0.320	259
CRM with one answer types	TREC-provided judgment set	0.547	165
	Our document/passage retrievers	0.305	264

Table 4: Trec-10 results on different grained size (Yen et al. 2006)

It is evident that a QA-System can greatly leverage on using SVMs for classification, either for question classification or answer classification.

There are also a few things to note when using SVMs. Given the fact that SVMs are a supervised machine learning algorithm, it is necessary to compile a fair amount of train data. This is usually time and resource consuming. Also, the algorithm can only be trained into known domain., meaningthat the corpus used to extract the information needs to be known as a precondition. This presents a problem when new data in a specific domain is introduced. Applying this approach into a new domain also raises a problem since the structure or lexical grammar might not be the same. Finally, the performance of SVMs can be greatly diminished if the corpus and/or features needed grow considerably. We still have to take into consideration that if it’s possible to have a well structured corpus and training data, then the use of SVMs can yield a good accuracy as well as speed.

CHAPTER III

EXPERIMENT

Introduction

Different methods to construct a QA-System were discussed previously in this paper. While each one of the previous approaches have their strengths and weaknesses, each one have been a huge step in QA-Systems. In the following section I will present my approach to QA-Systems using some of the methods discussed previously. The reason for this implementation is to present an application of the topics discussed in this paper.

Motivation

After close analysis of different approaches to implement a QA-System I noticed that probability models are a powerful tool to extract information from text and other data. Probabilistic models, like LSI (Oh et al. 2007), PLSA (Hofmann et al. 2001) or LDA (Blei et al. 2003), can help on making a better sense of the data that we are analyzing in order to return a desired answer. For these reasons I chose to use the LDA model to implement a Knowledge Base Community QA-System.

One of the purposes of this work is to suggest one of many paths to construct a truly open context QA-System. Probabilistic models help us in this task since they can infer meaningful information from text without an extensive knowledge of the language or the domain of the given question (Blei et al. 2003). For this reason the LDA model was chosen to implement a QA-

System. The system executes some lightweight Natural Language Processing (NLP) operations over the data. The NLP operations are kept to a minimum in order to infer as little as possible from the previous language knowledge of the programmer and language databases. Instead, the YahooAnswers database is used to extract candidate answers and the Bing WebAPI is used to extrapolate information from the given question. The architecture of this QA-System suggest the possibility to answer complex and simple questions based mainly on the probabilistic models and not on previous natural language knowledge.

This implementation was created taking into consideration the TREC-2015 track LiveQA. The track consists of a set of questions given to a QA-System, one every minute for a window of 24 hours. The questions database is constructed using YahooAnswers[4] information. The results are rank between 0 - 4 by members of TREC. The structure of the implemented QA-System is discussed below.

Structure of the QA-System

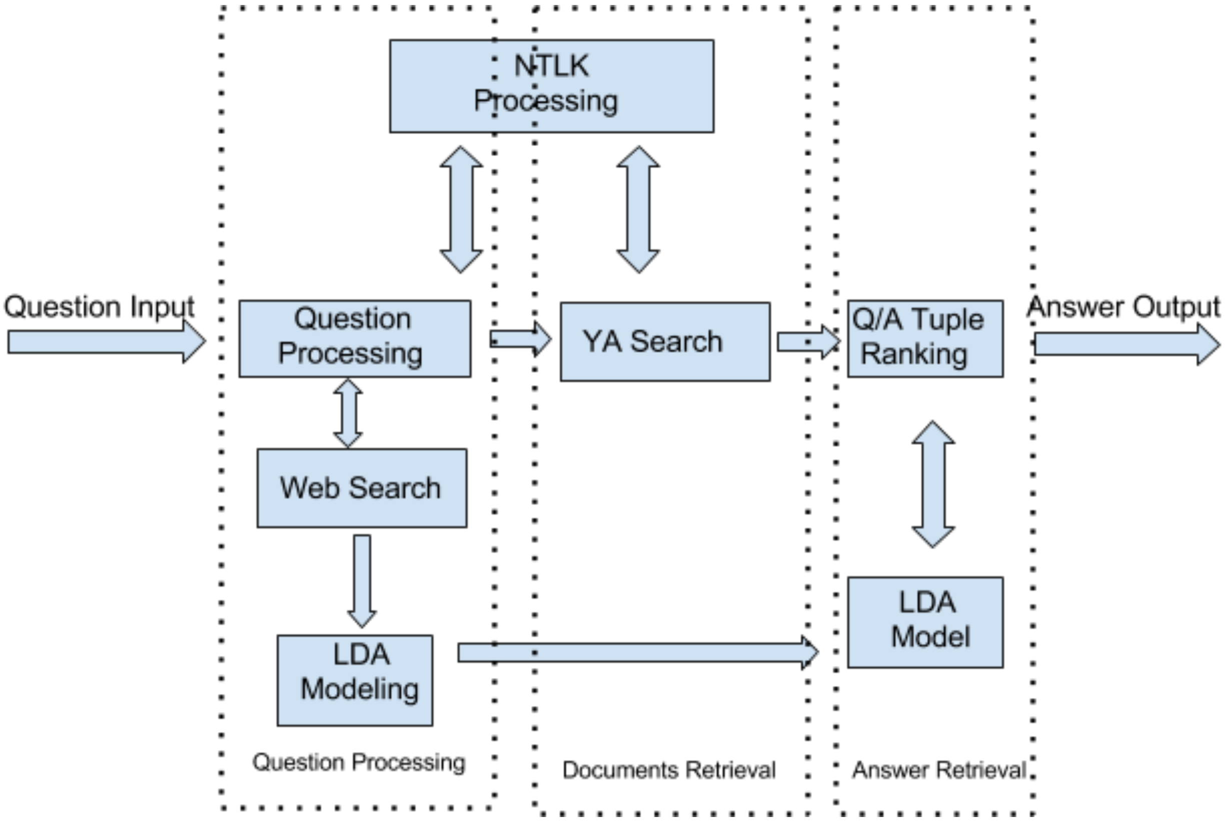


Figure 4. QA-System block diagram.

Figure 4 represents a block diagram of the QA-System architecture implemented in this paper. The "question processing" block uses the "NLTK processing" block to extract the keywords for further queries. The "NLTK processing" uses the python NLTK library to remove stop words, punctuation and then uses a lemmatizer to normalize the keywords. This process takes place on every document in the corpus constructed. The "Web Search" block uses the Bing Web API and the previously processed question to extract document related to the given query. The "YahooAnswers Search" block is used to retrieve candidate related question/answer tuples from the YahooAnswers database. The "LDA Modeling" block uses the gensim python library to construct a model based on LDA (Blei et al. 2003). Gensim implements an online LDA model, meaning that the probabilistic variables are calculated with each iteration of the algorithm. The

Answer ranking model feeds every candidate question/answer tuple into the "LDA Modeling" to get a topic probabilistic distribution of each of the question/answer tuples and then calculates the Jensen-Shannon distance (JSD) to get a similarity measure between the given question and the candidates question/answer tuples. Finally, the answer from the question/answer tuple with the shortest JSD to the given question is selected as the answer.

There are a few inferences made based on a quick analysis of the YahooAnswers database. First, it is observed that the title of each question includes the most important keywords referent to that question. This observation is used when searching for related documents using the Bing Web API. Second, it is noted that the first word usually describe the type of question that is being asked. For instance *"Which teams are going to participate in March Madness?"*, they word *"Which"* together with the rest of the keywords ("team", "go", "participate", "march", "madness") yields better results when searching for related documents in the web. Third, when searching for candidate questions in YahooAnswers and using a large amount of keywords, the results are not very good and often times the only result is the given question itself. In order to get better results when searching YahooAnswers multiple searches are conducted by randomly removing keywords until a the results yield at least 10 candidate related question/answer tuples.

The steps of the QA-System implemented in this paper are the following. The first block of the system is in charge of the question processing. First, when processing a given question the system retrieves the title and body of the question. Second, the question is process to extract the keywords by removing stop words, punctuation and lemmatizing the keywords using the NLTK package with the WordNet database. After extracting the keywords, these are fed into the Bing Web Search API and a set of 20 documents are processed the same way the question is processed. Then, an LDA model is constructed to infer the topics the given question contains. This is done using the python package gensim.

Second, The keywords extracted from the question are then used as a query to the YahooAnswers service to retrieve a set of 50 candidate related questions. The retrieved questions and answers are used to construct the corpus, for this they are processed by removing stop words, punctuation and lemmatizing the keywords.

Finally, the LDA model is used to calculate the probability distribution of the topics for each document in the corpus. Then, the Jenssen-Shannon distance (JSD) divergence is calculated for each of the documents. The JSD is calculated as a similarity measure between pairs of question/answers from the corpus and the given question. The candidate related questions are then ranked from more related to less related. The system then returns the more related pair of candidate questions/answers pairs. With the top related candidate question/answer pair the more upvoted answer is then return as the result. The JSD was used a similarity measure based on the work of Celikyilmaz et al. (2010). The JSD measures the shannon entropy between two probability measures. The JSD is often used instead of the KL-Divergance because it is symmetric and, as such, a true metric.

CHAPTER IV

RESULTS

A set of 200 questions were extracted from YahooAnswers at random. The questions are questions already answered, this is to better judge the accuracy of the system. Each answer for each given question is ranked by a human judging by the similarity between the result answer and the actual answer from the given question. A Mean Reciprocal Rank function is used to present the results as well as a score between 0 - 4 for each question where 0 - unreadable/no answer, 1 - poor, 2 - fair, 3 - good, 4 - excellent. The score system is based on the TREC 2015 LiveQA System.

	MRR	Score Avg.
LDA QA-System	0.4756	0.574

Table 5: Mean Reciprocal Ranking result and Score Average of 1000 random YahooAnswers questions judged by 5 humans.

The results in Table 5 are very promising, judging by the results on the TREC 2015 LiveQA. The score average recorded by TREC in the last year was 0.467 using 1087 questions judged by members of the TREC committee.

An example of the implemented QA-System is depicted below.

Input question:

"*Am I expected to pay for parking for everyone if my son has his birthday at the zoo? I plan on paying admission.?.*"

The system first processes the question title with the help of the NLTK library. The result of this processing is: "*Am paying everyone pay admission son birthday plan parking expected zoo*". This set of keywords are then used as a Web Search query in order to extract 20 related documents. The retrieved documents and the title and body of the question are put together to construct a corpus and fit the LDA model.

Then, a set of 20 topics are constructed from the corpus. The topics with a sample of 10 words constructed from the corpus:

```
(0, u'0.018*guest + 0.017*zoo + 0.016*student + 0.013*ride + 0.013*school + 0.010*pas + 0.010*money + 0.009*program +
0.008*service + 0.008*ticket')
(1, u'0.038*ride + 0.032*review + 0.025*zoo + 0.015*guest + 0.013*ticket + 0.012*pas + 0.012*student + 0.011*check +
0.011*season + 0.010*class')
(2, u'0.054*party + 0.035*guest + 0.022*invitation + 0.020*ticket + 0.016*gift + 0.010*student + 0.009*information +
0.009*ride + 0.009*present + 0.009*need')
(3, u'0.028*student + 0.021*school + 0.018*zoo + 0.014*money + 0.013*ride + 0.011*guest + 0.010*cost + 0.010*ticket +
0.008*program + 0.007*available')
(4, u'0.028*zoo + 0.019*party + 0.015*gift + 0.014*ticket + 0.013*student + 0.012*review + 0.012*invitation + 0.011*guest +
0.011*school + 0.010*money')
(5, u'0.035*ticket + 0.025*answer + 0.019*report + 0.016*comment + 0.015*student + 0.012*ride + 0.010*april + 0.010*think +
0.009*question + 0.009*page')
(6, u'0.030*ticket + 0.029*guest + 0.023*ride + 0.014*pas + 0.012*season + 0.011*visit + 0.011*zoo + 0.009*review +
0.009*service + 0.009*check')
(7, u'0.079*zoo + 0.018*answer + 0.016*comment + 0.016*pm + 0.014*report + 0.014*site + 0.010*question + 0.010*party +
0.009*state + 0.009*22')
(8, u'0.031*ride + 0.023*student + 0.019*guest + 0.017*school + 0.015*season + 0.014*pas + 0.013*money + 0.012*ticket +
0.009*available + 0.009*area')
(9, u'0.027*zoo + 0.022*report + 0.022*party + 0.022*answer + 0.017*comment + 0.017*student + 0.011*review + 0.010*think
+ 0.010*guest + 0.009*school')
(10, u'0.025*zoo + 0.017*review + 0.015*ticket + 0.012*ride + 0.011*student + 0.010*school + 0.009*state + 0.008*check +
0.007*guest + 0.007*love')
(11, u'0.043*zoo + 0.018*review + 0.016*ride + 0.014*student + 0.010*school + 0.010*ticket + 0.010*00 + 0.009*class +
0.008*train + 0.008*pas')
(12, u'0.030*zoo + 0.020*review + 0.018*student + 0.015*00 + 0.015*ride + 0.010*school + 0.008*money + 0.008*pas +
0.008*visit + 0.008*area')
(13, u'0.046*zoo + 0.043*review + 0.015*party + 0.011*guest + 0.010*ride + 0.010*gift + 0.010*attraction + 0.009*00 +
0.009*check + 0.008*class')
(14, u'0.027*ticket + 0.026*pas + 0.026*guest + 0.024*ride + 0.020*report + 0.014*service + 0.013*party + 0.010*answer +
0.010*season + 0.009*visit')
(15, u'0.054*zoo + 0.024*ride + 0.023*review + 0.014*april + 0.013*00 + 0.011*event + 0.011*class + 0.011*area + 0.010*fl +
0.010*guest')
(16, u'0.053*zoo + 0.034*review + 0.019*student + 0.017*ticket + 0.013*school + 0.011*last + 0.009*ride + 0.009*guest +
0.008*attraction + 0.008*money')
(17, u'0.049*zoo + 0.030*review + 0.020*ride + 0.014*guest + 0.013*00 + 0.012*ticket + 0.010*check + 0.009*visit +
0.009*class + 0.008*pas')
(18, u'0.042*student + 0.029*school + 0.017*zoo + 0.016*money + 0.013*report + 0.010*ride + 0.009*answer + 0.009*cost +
0.009*guest + 0.009*paying')
(19, u'0.045*zoo + 0.015*review + 0.015*ride + 0.012*party + 0.011*school + 0.010*student + 0.010*money + 0.010*class +
0.009*gift + 0.009*00')
```

We can see how the topics successfully represent what the question is asking.

The set of keywords are then used to query the YahooAnswers service to retrieve at the most 50 candidate related question/answer tuples. With these set of question/answer tuples the system transforms each tuple using the previously fitted LDA model to get a probability distribution of the topics.

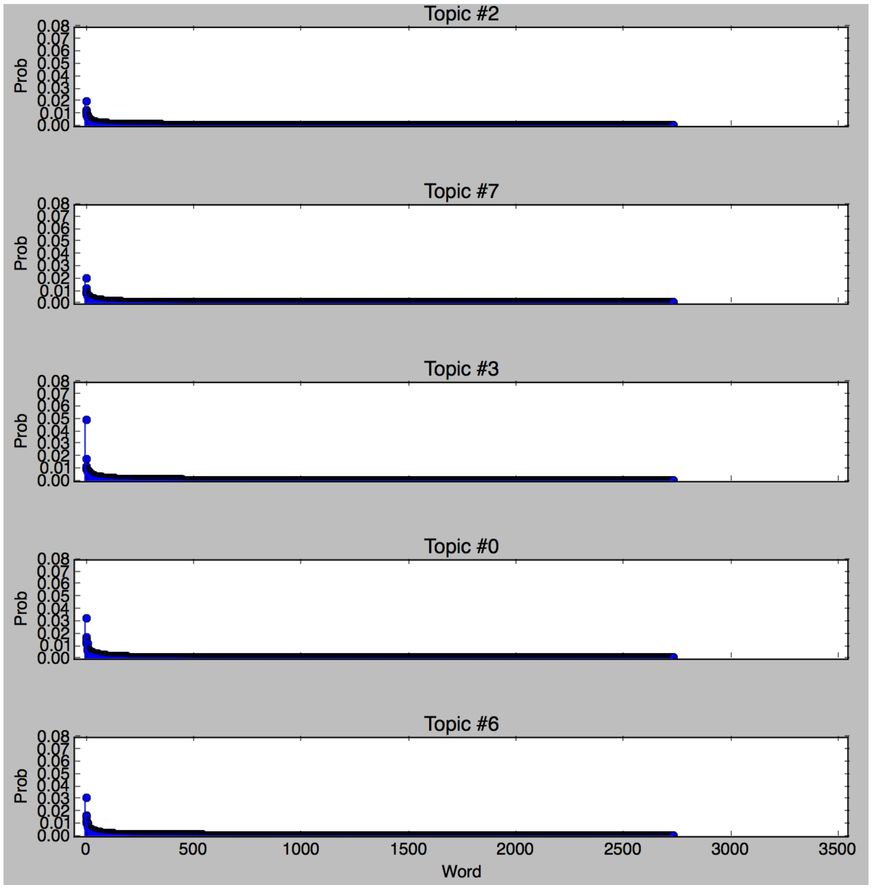


Figure 5 Word-topic distributions.

In Figure 5 the word-topic distributions is presented using all of the words in the corpus. The frequency of the words is used to construct the LDA model. The graph X axis in Figure 5 represent the word number related to 5 random topics. We can see that the distribution has some noise in it. In order to mitigate this the extremes of the word frequency array are removed. Meaning, the words that are repeated 90% with respect of the rest of the vocabulary and the words that are repeated less than 5% are removed.

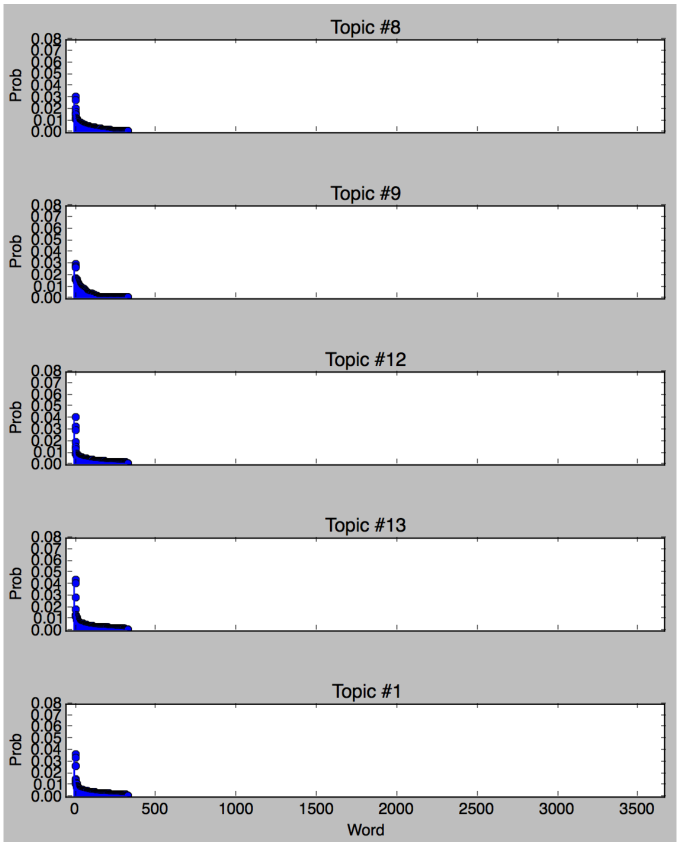


Figure 6 Word-topic distributions without extremes.

In Figure 6 the graphs for word-topic distribution are presented. Here the extremities of the word frequency counts are extracted. We can see that the topics become more exact as to which words constitute each topic.

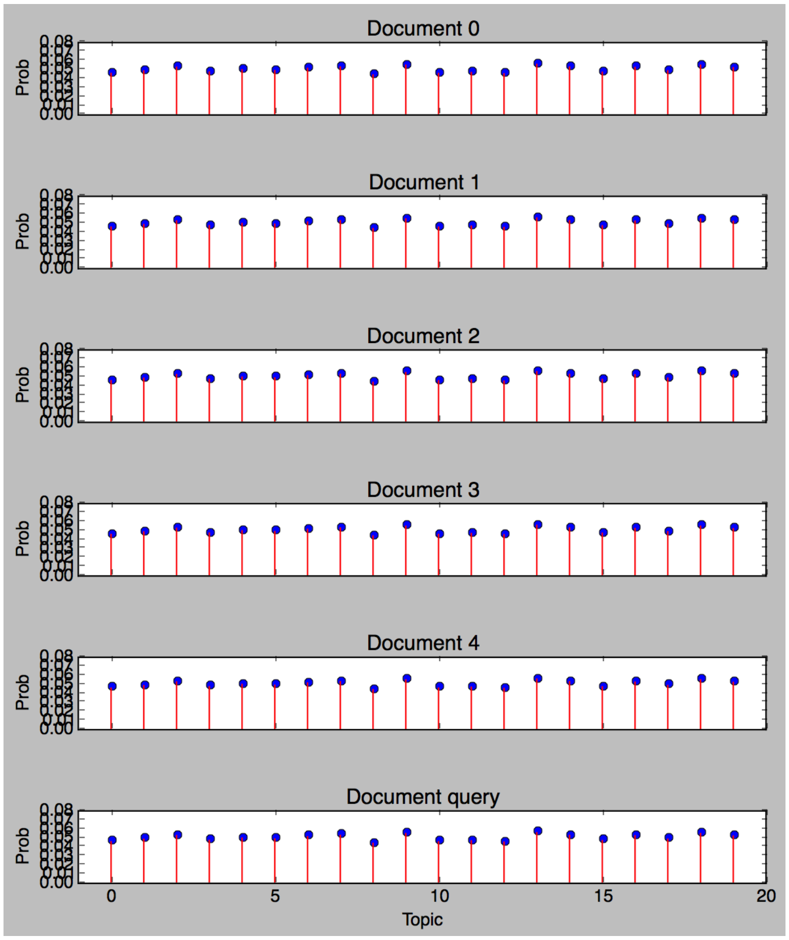


Figure 7 Document-topic probability distribution.

After the LDA model construction the system transforms each candidate query/answer tuple using the fitted LDA model. In Figure 8 we can see the document-topic probability distribution of the top 5 related question/answer tuples and the document-topic probability distribution of the question given as input. It is evident that their document-topic probability distribution is very similar. As a similarity metric the Jensen-Shannon Distance is calculated between the document-topic probability distribution of the question given as input and all of the 50 candidate question/answer tuples retrieved from the YahooAnswer service. The system then ranks the candidate question/answer tuples by their JSD with the given question and returns the top answer as a result. In this example the answer given as a result is:

“*Save the Zoo party when your little one is 7 or 8. It sounds like fun having a costume party at the Zoo. Keep in mind, the rule is to invite children to your sons party that he has been close to this past year. Maybe look into having the party at your local Pizza place. I did this when my children were young, it was great the Pizza Hut had the cake, drinks, pizza and party decorations for a minimal cost. Best part they cleaned up. Rule of thumb, what ever the age of the child invite that many children. At 7 invite 7 friends and so on. For his first birthday give him a small cake for him to dig into all by himself. Get plenty of pictures. My birthday is on Halloween, my memories are going trick or treating/costume parties. I am half a Century plus one, soon to be two this year, I stay home to treat the Little goblins. I love that time of year. Just have fun, he will have many more to remember. Good luck!*”

We can see that the system was able to identify the similarity between the candidate question/answer tuples and the given question. The other top 4 answers retrieve also talk about children birthday parties but they also contain other topics such as family not getting along, who to invite, or mistakes made at birthday parties. The given answer also corresponds to another, similar question: “*Mom etiquette question, birthday party!?*”. Although the question corresponding to the given answer is different and shorter that the given question we can see that the answer does give some of the information that the input was asking about. This answer can be ranked as 1 or 2 given the fact that the answer does not give information about paying the parking ticket but it does suggest another idea which can be helpful.

CHAPTER V

CONCLUSIONS

Information retrieval is the main topic of Question Answering. This information retrieval usually takes place on natural language documents. Because of the nature of natural language, retrieving information from a document has been proven to be very difficult. If we take a small step back and look at question answering outside of the computer science domain, it is evident that even human beings struggle with this task, not only when trying to respond to a question but also when trying to construct questions. In computer science many people have been trying to mitigate these problems by doing more deep analysis of the information and the input question as well as developing more robust algorithms to classify the given data.

In this thesis I talked about some of the different approaches proposed in the past, take into consideration that there's still a lot of research been done in this, and adjacent, subject. Each of the previously laid out methods have their strengths and weaknesses. Their performance depends on different variables, e.g. if enough training data exists, if the existent data is correctly labeled or tagged or if the data conforms a coherent structure, just to mention a few of these variables. I presented information which conveys the success of each of the different approaches within specific domains or given a specific set of preconditions. This also illustrated that we, as a community, are advancing forward in solving this problem. None the less, this can still be categorized as an open problem and improvement can be done.

It is imperative to mention that there is a need to keep improving or creating new methods to tackle this problem. As users and human beings, we will always have questions that need answers and we have to understand that the answer is as important as the question itself. We have access to an incredible amount of data, but data by itself is of no use. There is a need to make sense of the vast amount of data available. By successfully doing this we could greatly improve the way we interact with machines and, probably, with each other as well as how we evolve intellectually. Think about how fast could a research advance if we had a more advanced and robust tool to analyze data and extract desired information. What about facilitating decision making by being able to retrieve answers to key questions about a subject in a promptly manner?. We might even learn more about ourselves if we could analyze all the data we, as a person, creates in an accurate and available way.

Experiment

The experiment result is very promising when compared to the TREC 2015 LiveQA track. These results suggest that a QA-System implementation consisting of mostly probabilistic models can perform efficiently to answer complex questions. There are some drawbacks that are necessary to address. First, it was needed to draw some inferences judging by a the used knowledge database in order to boost the performance of the system. Nonetheless these inferences were kept to a minimum to present an unsupervised learning approach to question answering. Also, the results suggest that a probability model approach is satisfactory when automatically answering questions of different domain with very little previous knowledge of the domain. Second, being a knowledge based QA-System the answers given by the system are only as good as the knowledge database.

Future Work

The focus of this thesis was to present different approaches to Question Answering Systems (QA-Systems) and analyze them presenting their strengths and weaknesses as well as talking about the future work that could be done in this domain. A QA-System based on an LDA modeling approach was implemented to suggest a step further into an open domain QA-System.

The LDA modeling approach was first proposed by (Blei et al. 2003). It is important to note that the ideas used in this paper do not represent the end of the road for an open domain QA-System. In the future other probabilistic modeling approaches will be explored as well as different similarity measures.

As we can see a large amount of work has been done towards QA-Systems using different approaches. This is mainly because of all the different variables that we must take into consideration when answering a question. Not only about the data we have available to extract the information but also how the question is presented as well as how the answer should be returned.

Every day better and more robust algorithms and models to analyze lexical information are been created. I believe machine learning algorithms can be the answer to an open domain QA-System. Possibly a mixture of multiple of these approaches. This because of the capacity of these algorithms to learn from data. I think the next step is for a machine learning algorithm to be able to continuously learn from data as well as from the interaction between the algorithm and its users, between users or even between the algorithm and itself. This is because we are trying to extract information from human created data, and human beings are constantly changing the way they think. For this simple idea, an evolutionary-learning algorithm might be the path to walk to an open domain QA-System.

There is still other aspects of a QA-System that could evolve in the near future. Question decomposition has been proven to be an efficient approach to question answering. The reason why is obvious and human beings do it when asking or answering questions between each other. I believe a QA-System can benefit from more question pre-processing facets. Sometimes a user does not know what question to ask or how to ask it. If we are able to have a back and forth conversation with a QA-System this might become easier. Of course this idea involves even more complex topics, but it is still worth mentioning.

Finally, we can talk about the answer construction. In this paper I explained and analyzed multiple approaches to construct answers either from extracted sentences, extracted passages or abstracted information. We have to keep in mind that all of this processing is being done for a user and user feedback is always very important when presenting any type of information. I believe answer construction should not only use the data in the corpus but also data given by the user. There can be a lexical analysis on how the input question was constructed or maybe the QA-System could have access to some of the user's data. With this the QA-System could tailor the answer specifically for that user and present the information in terms that the user can easily understand. This could be a great advantage for humanity since learning anything could be made very easy.

Overall there has been a lot of improvement in this domain and we are moving forward at a very fast pace. QA-Systems and information retrieval is been used by a lot of systems that we use daily and they do make our life easier.

REFERENCES

Adel Tahri and Okba Tibermacine *DBpedia based factoid question answering system*. International Journal of Web & Semantic Technology (IJWesT) Vol.4, No.3, July 2013

Asli Celikyilmaz , Dilek Hakkani-tur , Gokhan Tur. Lda based similarity modeling for question answering. In Proceedings of the NAACL HLT 2010 Workshop on Semantic Search.

Blei David, Ng Andrew, Jordan I. Michael. Latent Dirichlet Allocation. Journal of Machine Learning Research 3 (2003) 993-1022

Carbonell, J. and J. Goldstein. 1998. *The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries*. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998), pages 335-336, Melbourne, Australia.

C. Nobata, S. Sekine, M. Murata, K. Uchimoto, M. Utiyama, and H. Isahara. 2001. Sentence Extraction System Assembling Multiple Evidence. Proc. of the 2nd NTCIR Workshop, pages 319–324.

Daniel Z, Ján S, Jozef J, Anton C. Text Categorization with Latent Dirichlet Allocation. Journal of Electrical and Electronics Engineering. 2014;7(1):161-4.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty (2010) Building Watson: An Overview of the DeepQA Project. Association for the Advancement of Artificial Intelligence, Fall 2010.

Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R. Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science (1986-1998). 1990;41(6):391.

Deborah L. McGuinness, Question Answering on the Semantic Web. IEEE Intelligent Systems Volume:19 ,Issue: 1

Diekema, A.R., Yilmazel, O., Chen, J., Harwell, S., He, L., and Liddy, E.D. Finding Answers to Complex Questions. In Maybury, M.T. (Ed.) New Directions in Question Answering. The MIT Press, 2004, p. 141-152.

Donna Harman Overview of the First Text REtrieval Conference (TREC-1). National Institute of Standards and Technology Gaithersburg, Md. 20899

Erkan, G. and D. R. Radev. 2004. *LexRank: Graph-based Lexical Centrality as Salience in Text Summarization*. Journal of Artificial Intelligence Research, 22:457-479.

E. Brill. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging. Computational Linguistics, 21(4), 1995.

F.E. Chakik, A. Shahine, J. Jaam, A. Hasnah, An approach for constructing complex discriminating surfaces based on Bayesian interference of the maximum entropy, Information Sciences 163 (4) (2004) 275–291

Garrison JA. UpToDate. *Journal of the Medical Library Association*. 2003;91(1):97.

Hirao, T., H. Isozaki, E. Maeda, and Y. Matsumoto. 2002a. Extracting Important Sentences with Support Vector Machines. In Proceedings of the 19th International Conference on Computational Linguistics, pages 1-7, Taipei, Taiwan.

Harabagiu, A. Hickl, J. Lehmann, and D. Moldovan. Experiments with Interactive Question-Answering. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), 2005.

Hyo-Jung Oh, Sung Hyon Myaeng, Myung-Gil Jang. Semantic passage segmentation based on sentence topics for question answering. *Information Sciences 177 (2007) 3696–3717*

Hofmann T. Unsupervised Learning by Probabilistic Latent Semantic Analysis. Machine Learning. 2001;42(1):177-96.

H. Kim, K.Kim, G.G.Lee,and J.Seo. MAYA: A fast question-answering system based on a predictive answer indexer. In *Proceedings of the Association for Computational Lin guistics 39th Annual Meeting and 10th Conference of the European Chapter Workshop on Open-Domain Question Answering,* page 916, 2001.

Kang Liu, Jun Zhao, Shizhu He, and Yuanzhe Zhang, Question Answering Over Knowledge Bases. Institute of Automation, Chinese Academy of Sciences, 2015.

K. Zechner. 1996. Fast Generation of Abstracts from General Domain Text Corpora by Extracting Relevant Sentences. Proc. of the 16th COLING, pages 986–989.

K. Hacioglu, S. Pradhan, W. Ward, J. H. Martin, and D. Jurafsky. Shallow semantic parsing using support vector machines. In *Proceedings of the Human Language Technology Conference/North American chapter of the Association for Computational Linguistics annual meeting (HLT/NAACL-2004),*2004. URL http://www.stanford.edu/ \~{}jurafsky/hlt-2004-verb.pdf.

Linker S. A Knowledge Base and Question Answering System Based on Loglan and English [dissertation]. ProQuest Dissertations Publishing; 2011.

Liu T, Zhang W, Cao L, Zhang Y. Question Popularity Analysis and Prediction in Community Question Answering Services: e85236. PLoS One. 2014;9(5).

Li X, Ouyang J, Zhou X, Lu Y, Liu Y. Supervised labeled latent Dirichlet allocation for document categorization. Applied Intelligence. 2015;2014;42(3):581.

Lienou M, Maitre H, Datcu M. Semantic Annotation of Satellite Images Using Latent Dirichlet Allocation. IEEE Geoscience and Remote Sensing Letters. 2010;7(1):28-32.

M. Collins. Head-Driven Statistical Models for Natural Language Parsing. PhD thesis, University of Pennsylvania, 1999.

M. Hearst, Multi-paragraph segmentation of expository text, in: Proceedings of the 32nd Annual meeting of the Association of Computational Linguistics (ACL-94), 1994, pp. 9–16.

Otterbacher, J., G. Erkan, and D. R. Radev. 2005. Using Random Walks for Question focused Sentence Retrieval. *In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 915-922, Vancouver, Canada.

Preeti and Brahmaleen Kaur Sidhu (2013) NATURAL LANGUAGE PROCESSING. A Vinitha et al, Int.J.Computer Technology & Applications,Vol 4 (5),751-758

Rafael M. Terol, Patricio Martínez-Barco, Manuel Palomar *A knowledge based method for the medical question answering problem* Computers in Biology and Medicine 37 (2007) 1511 – 1521

Rasiwasia N, Vasconcelos N. Latent Dirichlet Allocation Models for Image Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2013;35(11):2665-79.

Sofia J. Athenikos, Hyoil Han *Biomedical question answering: A survey*. computer methods and programs in biomedicine 99 (2010) 1–24

Saquete, Vicedo, Mart ́ınez-Barco, Munoz, & Llorens *Enhancing QA Systems with Complex Temporal Question Processing Capabilities* Journal of Artificial Intelligence Research 35 (2009) 775-811

Sparck Jones, K. (1972), “A statistical interpretation of term specificity and its application in retrieval”, Journal of Documentation, Vol. 28, pp. 11–21.

S. Harabagiu, F. Lacatusu, and A. Hickl. Answering complex questions with random walk models. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pages 220 - 227. ACM, 2006.

S. Tellex, B. Katz, J.J. Lin, A. Fernandes, G. Marton, Quantitative evaluation of passage retrieval algorithms for question answering, in: Proceedings of the 26th ACM SIGIR Conference on Research and Development in Information Retrieval, 2003, pp. 41–47.

T. Brants, F. Chen, I. Tsochantaridis, Topic-based document segmentation with probabilistic latent semantic analysis, in: Proceedings of the 11th International Conference on Information and Knowledge management (CIKM-02), 2002, pp. 211–218.

T. Hirao, M. Hatayama, S. Yamada, and K. Takeuchi. 2001. Text Summarization based on Hanning Window and Dependency structure analysis. Proc. of the 2nd NTCIR Workshop, pages 349–354.

T. Nomoto and Y. Matsumoto. 1997. The Reliability of Human Coding and Effects on Automatic Abstracting (in Japanese). The Special Interest Group Notes of IPSJ (NL-120-11), pages 71–76.

West R, Gabrilovich E, Murphy K, Sun S, Gupta R, Lin D. Knowledge base completion via search-based question answering. ACM; 2014.

Yen, SJ, et al. "A Support Vector Machine-Based Context-Ranking Model for Question Answering." *INFORMATION SCIENCES* 224 (2013): 77-87. Web.

Yirdaw, E., & Ejigu, D. (2012). Topic-based amharic text summarization with probabilistic latent semantic analysis. Paper presented at the 8-15. doi:10.1145/2457276.2457279

Yong Gang Cao, Feifan Liu, Pippa Simpson, Lamont Antieau, Andrew Bennett,d, James J. Cimino, John Ely , and Hong Yua, (2011) AskHERMES: An online question answering system for complex clinical questions. J Biomed Inform. 2011 April ; 44(2): 277–288.

BIOGRAPHICAL SKETCH

Josue Balandrano Coronel was born in Cd. Victoria, Tamaulipas in 1985. He attended the Instituto Tecnológico de Monterrey from December 2004 to June 2008 receiving his Bachelor in Science in Electronic Systems Engineering. He worked in Balco Joyeros as a Full Stack Developer from 2008 to 2012. In 2013 he started working as a Web Software Engineer II at the University of Texas-Pan American in the Internet Services department until Jun 2015. In July 2015 he started working at the Texas Advanced Computer Center as a Research Engineer / Scientist Associate III. He received his Masters of Science in Computer Science in May 2016.