International Journal of Research and Reviews in Information Sciences (IJRRIS)
Vol. 2, No. 3, September 2012

www.sciacademypublisher.com

The Question Answering Systems: A Survey.

Ali Mohamed Nabil Allam[1] and Mohamed Hassan Haggag[2]

[1] College of Management & Technology, Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt.
[2] Faculty of Computers & Information, Helwan University, Cairo, Egypt.
Email: ali.allam@staff.aast.edu , m_h_haggag@yahoo.com

Abstract – Question Answering (QA) is a specialized area in the field of Information Retrieval (IR). The QA systems are concerned with providing relevant answers in response to questions proposed in natural language. QA is therefore composed of three distinct modules, each of which has a core component beside other supplementary components. These three core components are: question classification, information retrieval, and answer extraction. Question classification plays an essential role in QA systems by classifying the submitted question according to its type. Information retrieval is very important for question answering, because if no correct answers are present in a document, no further processing could be carried out to find an answer. Finally, answer extraction aims to retrieve the answer for a question asked by the user. This survey paper provides an overview of Question-Answering and its system architecture, as well as the previous related work comparing each research against the others with respect to the components that were covered and the approaches that were followed. At the end, the survey provides an analytical discussion of the proposed QA models, along with their main contributions, experimental results, and limitations.

Keywords – Question Answering, Natural Language Processing, Information Retrieval, Question Classification, Answer Extraction, Evaluation Metrics

1. Introduction

Question Answering (QA) is a research area that combines research from different, but related, fields which are Information Retrieval (IR), Information Extraction (IE) and Natural Language Processing (NLP).
Actually, what a current information retrieval system or search engine can do is just "document retrieval", i.e. given some keywords it only returns the relevant ranked documents that contain these keywords. Information retrieval systems do not return answers, and accordingly users are left to extract answers from the documents themselves. However, what a user really wants is often a precise answer to a question. [1], [2]. Hence, the main objective of all QA systems is to retrieve answers to questions rather than full documents or best-matching passages, as most information retrieval systems currently do.
However, the main type of questions submitted by users in natural language are the factoid questions, such as "When did the Egyptian revolution take place?" But, the recent research trend is shifting toward more complex types of questions such as definitional questions (e.g. "Who is the President of Egypt?" or "What is SCAF?"), list questions (e.g. "List the countries that won the Cup of African Nations"), and why-type questions (e.g. "Why was Sadat assassinated?").
The Text Retrieval Conference (TREC), a conference series co-sponsored by NIST, initiated the Question-Answering Track in 1999 which tested systems' ability to retrieve short text snippets in response to factoid questions (for example, "How many calories are in a Big Mac?") [3]. Following the success of TREC, in 2002 the workshops of both the Cross Language Evaluation Forum (CLEF) and NII Test Collection for IR Systems (NTCIR) started multilingual and cross-lingual QA tracks, focusing on European and Asian languages respectively [4].
Moreover, QA systems are classified into two main categories, namely open-domain QA systems and closed-domain QA systems. Open-domain question answering deals with questions about nearly everything and can only rely on universal ontology and information such as the World Wide Web. On the other hand, closed-domain question answering deals with questions under a specific domain (music, weather forecasting etc.) The domain specific QA system involves heavy use of natural language processing systems formalized by building a domain specific ontology. [5]

2. QA System Components

As shown in (Figure 1), a typical QA system consists of three distinct modules, each of which has a core component beside other supplementary components: "Query Processing Module" whose heart is the question classification, the "Document Processing Module" whose heart is the information retrieval, and the "Answer Processing Module" whose heart is the answer extraction.
Question processing is the module which identifies the focus of the question, classifies the question type, derives the expected answer type, and reformulates the question into semantically equivalent multiple questions.
Reformulation of a question into similar meaning questions is also known as query expansion and it boosts up the recall of the information retrieval system. Information retrieval (IR) system recall is very important for question answering, because if no correct answers are present in a document, no further processing could be carried out to find an answer [6].

Precision and ranking of candidate passages can also affect question answering performance in the IR phase.

Answer extraction is the final component in question answering system, which is a distinguishing feature between question answering systems and the usual sense of text retrieval systems. Answer extraction technology becomes an influential and decisive factor on question answering system for the final results. Therefore, the answer extraction technology is deemed to be a module in the question answering system [5].

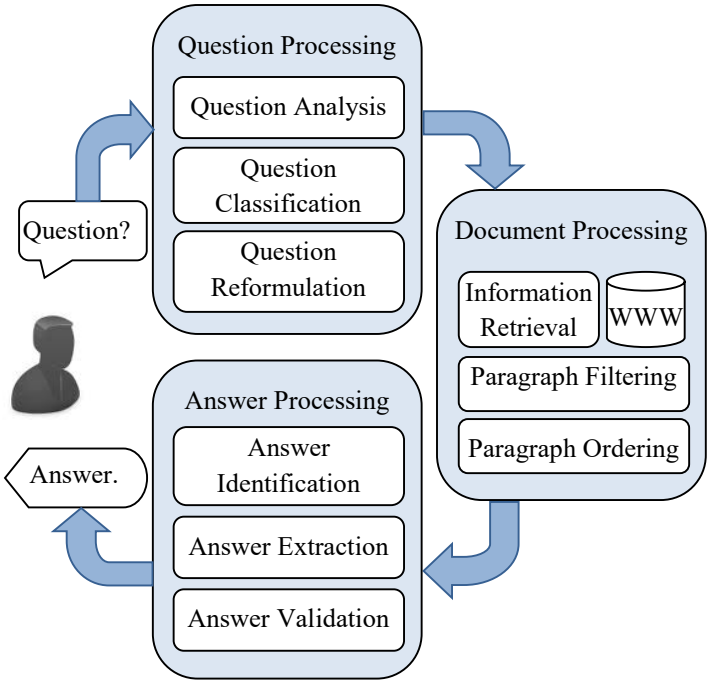

Figure 1: Question Answering System Architecture

Typically, the following scenario occurs in the QA system:

1. First, the user posts a question to the QA system.
2. Next the question analyzer determines the focus of the question in order to enhance the accuracy of the QA system.
3. Question classification plays a vital role in the QA system by identifying the question type and consequently the type of the expected answer.
4. In question reformulation, the question is rephrased by expanding the query and passing it the information retrieval system.
5. The information retrieval component is used to retrieve the relevant documents based upon important keywords appearing in the question.
6. The retrieved relevant documents are filtered and shortened into paragraphs that are expected to contain the answer.
7. Then, these filtered paragraphs are ordered and passed to the answer processing module.
8. Based on the answer type and other recognition techniques, the candidate answers are identified.
9. A set of heuristics is defined in order to extract only the relevant word or phrase that answers the question.
10. The extracted answer is finally validated for its correctness and presented to the user.

2.1 Question Processing Module

Given a natural language question as input, the overall function of the question processing module is to analyze and process the question by creating some representation of the information requested. Therefore, the question processing module is required to:

- ***Analyze*** the question, in order to represent the main information that is required to answer the user's question.
- ***Classify*** the question type, usually based on taxonomy of possible questions already coded into the system, which in turn leads to the expected answer type, through some shallow semantic processing of the question.
- ***Reformulate*** the question, in order to enhance the question phrasing and to transform the question into queries for the information retrieval (search engine).

These steps allow the question processing module to finally pass a set of query terms to the document processing module, which uses them to perform the information retrieval.

2.1.1 Question Analysis

Question analysis is also referred to as "Question Focus". Unfortunately, classifying the question and knowing its type is not enough for finding answers to all questions. The "what" questions in particular can be quite ambiguous in terms of the information asked by the question [7]. In order to address this ambiguity, an additional component which analyzes the question and identifies its ***focus*** is necessary.

The focus of a question has been defined by Moldovan *et al.* [8] to be a word or sequence of words which indicate what information is being asked for in the question. For instance, the question "*What is the longest river in New South Wales?*" has the focus *"longest river"*. If both the question type (from the question classification component) and the focus are known, the system is able to more easily determine the type of answer required.

Identifying the focus can be done using pattern matching rules, based on the question type classification.

2.1.2 Question Type Classification

In order to correctly answer a question, it is required to understand what type of information the question asks for, because knowing the type of a question can provide constraints on what constitutes relevant data (the answer), which helps other modules to correctly locate and verify an answer.

The question type classification component is therefore a useful, if not essential, component in a QA system as it provides significant guidance about the nature of the required answer. Therefore, the question is first classified by its type: *what, why, who, how, when, where* questions, etc.

2.1.3 Answer Type Classification

Answer type classification is a subsequent and related component to question classification. It is based on a mapping of the question classification. Once a question has been classified, a simple rule based mapping would be used to determine the potential answer types. Again, because question classification can be ambiguous, the system should allow for multiple answer types.

2.1.4 Question Reformulation

Once the *"focus"* and *"question type"* are identified, the module forms a list of keywords to be passed to the information retrieval component in the document processing module. The process of extracting keywords could be performed with the aid of standard techniques such as named-entity recognition, stop-word lists, and part-of-speech taggers, etc.

Other methods of expanding the set of question keywords could include using an online lexical resource such as the WordNet ontology. The synsets (synonym sets) in WordNet could be used to expand the set of question keywords with semantically related words that might also occur in documents containing the correct question answer [9].

2.2 Document Processing Module

The document processing module in QA systems is also commonly referred to as paragraph indexing module, where the reformulated question is submitted to the information retrieval system, which in turn retrieves a ranked list of relevant documents. The document processing module usually relies on one or more information retrieval systems to gather information from a collection of document corpora which almost always involves the World Wide Web as at least one of these corpora [7]. The documents returned by the information retrieval system is then filtered and ordered. Therefore, the main goal of the document processing module is to create a set of candidate ordered paragraphs that contain the answer(s), and in order to achieve this goal, the document processing module is required to:

- ***Retrieve*** a set of ranked documents that are relevant to the submitted question.
- ***Filter*** the documents returned by the retrieval system, in order to reduce the number of candidate documents, as well as the amount of candidate text in each document.
- ***Order*** the candidate paragraphs to get a set of ranked paragraphs according to a plausibility degree of containing the correct answer.

The motivation for shortening documents into paragraphs is making a faster system. The response time of a QA system is very important due to the interactive nature of question answering. This ensures that a reasonable number of paragraphs are passed on to the answer processing module.

2.2.1 Information Retrieval (IR)

Information domains, such as the web, have enormous information content. Therefore, the goal of the information retrieval system is to retrieve accurate results in response to a query submitted by the user, and to rank these results according to their relevancy.

One thing to be considered is that it is not desirable in QA systems to rely on IR systems which use the cosine vector space model for measuring similarity between documents and queries. This is mainly because a QA system usually wants documents to be retrieved only when all keywords are present in the document. This is because the keywords have been carefully selected and reformulated by the Question Processing module. IR systems based on cosine similarity often return documents even if not all keywords are present.

Information retrieval systems are usually evaluated based on two metrics – precision and recall. Precision refers to the ratio of relevant documents returned to the total number of documents returned. Recall refers to the number of relevant documents returned out of the total number of relevant documents available in the document collection being searched. In general, the aim for information retrieval systems is to optimize both precision and recall. For question answering, however, the focus is subtly different. Because a QA system performs post processing on the documents returned, the recall of the IR system is significantly more important than its precision [7].

2.2.2 Paragraph Filtering

As mentioned before, the number of documents returned by the information retrieval system may be very large. Paragraph filtering can be used to reduce the number of candidate documents, and to reduce the amount of candidate text from each document. The concept of paragraph filtering is based on the principle that the most relevant documents should contain the question keywords in a few neighboring paragraphs, rather than dispersed over the entire document. Therefore, if the keywords are all found in some set of ***N*** consecutive paragraphs, then that set of paragraphs will be returned, otherwise, the document is discarded from further processing.

2.2.3 Paragraph Ordering

The aim of paragraph ordering is to rank the paragraphs according to a plausibility degree of containing the correct answer. Paragraph ordering is performed using ***standard radix sort*** algorithm. The radix sort involves three different scores to order paragraphs:

i. ***Same word sequence score:*** the number of words from the question that are recognized in the same sequence within the current paragraph window.
ii. ***Distance score:*** the number of words that separate the most distant keywords in the current paragraph window;
iii. ***Missing keyword score:*** the number of unmatched keywords in the current paragraph window.

A paragraph window is defined as the minimal span of text required to capture each maximally inclusive set of question keywords within each paragraph. Radix sorting is performed for each paragraph window across all paragraphs.

2.3 Answer Processing Module

As the final phase in the QA architecture, the answer processing module is responsible for identifying, extracting and validating answers from the set of ordered paragraphs passed to it from the document processing module. Hence, the answer processing module is required to:

- ***Identify*** the answer candidates within the filtered ordered paragraphs through parsing.
- ***Extract*** the answer by choosing only the word or phrase that answers the submitted question through a set of heuristics.
- ***Validate*** the answer by providing confidence in the correctness of the answer.

2.3.1 Answer Identification

The answer type which was determined during question processing is crucial to the identification of the answer. Since usually the answer type is not explicit in the question or the answer, it is necessary to rely on a parser to recognize named entities (e.g. names of persons and organizations, monetary units, dates, etc.). Also, using a part-of-speech tagger (e.g., Brill tagger) can help to enable recognition of answer candidates within identified paragraphs. The recognition of the answer type returned by the parser creates a candidate answer. The extraction of the answer and its validation are based on a set of heuristics [8].

2.3.2 Answer Extraction

The parser enables the recognition of the answer candidates in the paragraphs. So, once an answer candidate has been identified, a set of heuristics is applied in order to extract only the relevant word or phrase that answers the question.
Researchers have presented miscellaneous heuristic measures to extract the correct answer from the answer candidates. Extraction can be based on measures of distance between keywords, numbers of keywords matched and other similar heuristic metrics. Commonly, if no match is found, QA systems would fallback to delivering the best ranked paragraph. Unfortunately, given the tightening requirements of the TREC QA track, such behavior is no longer useful. As in the original TREC QA tracks, systems could present a list of several answers, and were ranked based on where the correct answer appeared in the list. From 1999-2001, the length of this list was 5. Since 2002, systems have been required to present only a single answer [10].

2.3.3 Answer Validation

Confidence in the correctness of an answer can be increased in a number of ways. One way is to use a lexical resource like WordNet to validate that a candidate response was of the correct answer type. Also, specific knowledge sources can also be used as a second opinion to check answers to questions within specific domains. This allows candidate answers to be sanity checked before being presented to a user. If a specific knowledge source has been used to actually retrieve the answer, then general web search can also be used to sanity check answers. The principle relied on here is that the number of documents that can be retrieved from the web in which the question and the answer co-occur can be considered a significant clue of the validity of the answer. Several people have investigated using the redundancy of the web to validate answers based on frequency counts of question answer collocation, and found it to be surprisingly effective. Given its simplicity, this makes it an attractive technique.

3. Literature Review

Historically, the best-known early question answering system was BASEBALL, a program developed by Green *et al.* [11] in 1961 for answering questions about baseball games played in the American league over one season. Also, the most well-remembered other early work in this field is the LUNAR system [12], which was designed in 1971 as a result of the Apollo moon mission, to enable lunar geologists to conveniently access, compare and evaluate the chemical analysis data on lunar rock and soil composition that was accumulating. Many other early QA systems such as SYNTHEX, LIFER, and PLANES [13] aimed to achieve the same objective of getting an answer for a question asked in natural language.

However, QA systems have developed over the past few decades until they reached the structure that we have nowadays. QA systems, as mentioned before, have a backbone composed of three main parts: question classification, information retrieval, and answer extraction. Therefore, each of these three components attracted the attention of QA researchers.

- **Question Classification:**

Questions generally conform to predictable language patterns and therefore are classified based on taxonomies. Taxonomies are distinguished into two main types: flat and hierarchical taxonomies. Flat taxonomies have only one level of classes without having sub-classes, whereas hierarchical taxonomies have multi-level classes. Lehnert [14] proposed "QUALM", a computer program that uses a conceptual taxonomy of thirteen conceptual classes. Radev *et al.* [15] proposed a QA system called NSIR, pronounced "answer", which used a flat taxonomy with seventeen classes, shown in (Table 1).

Table 1: Flat Taxonomy (Radev *et al.* – "NSIR")

PERSON	PLACE	DATE
NUMBER	DEFINITION	ORGANIZATIO
DESCRIPTIO	ABBREVIATIO	KNOWNFOR
RATE	LENGTH	MONEY
REASON	DURATION	PURPOSE
NOMINAL	OTHER	

In the proceedings of TREC-8 [10], Moldovan *et al.* [8] proposed a hierarchical taxonomy (Table 2) that classified the question types into nine classes, each of which was divided into a number of subclasses. These question classes and subclasses covered all the 200 questions in the corpus of TREC-8.

Table 2: Hierarchical Taxonomy (Moldovan *et al.*, TREC8)

Question class	Question	Answer Type
WHAT	basic-what	Money / Number / Definition / Title / NNP / Undefined
	what-who	
	what-when	
	what-where	
WHO		Person /
HOW	basic-how	Manner
	how-many	Number
	how-long	Time / Distance
	how-much	Money / Price
	how-much	Undefined
	how-far	Distance
	how-tall	Number
	how-rich	Undefined
	how-large	Number
WHERE		Location
WHEN		Date
WHICH	which-who	Person
	which-where	Location
	which-when	Date
	which-what	NNP /
NAME	name-who	Person /
	name-where	Location
	name-what	Title / NNP
WHY		Reason
WHOM		Person /

Harabagiu *et al.* [16] used a taxonomy in which some categories were connected to several word classes in the WordNet ontology. More recently, in the proceedings of TREC-10 [10], Li and Roth [17] proposed a two-layered taxonomy, shown in Table 3, which had six super (coarse) classes and fifty fine classes.

Table 3: Hierarchical Taxonomy (Li & Roth, TREC-10)

ABBREVIATION	Letter	Description	**NUMERIC**
Abbreviation	Other	Manner	Code
Expression	Plant	Reason	Count
ENTITY	Product	**HUMAN**	Date
Animal	Religion	Group	Distance
Body	Sport	Individual	Money
Color	Substance	Title	Order
Creative	Symbol	Description	Other
Currency	Technique	**LOCATION**	Period
disease medicine	Term	City	Percent
Event	Vehicle	Country	Size
Food	Word	Mountain	Speed
Instrument	**DESC**	Other	Temp
Language	Definition	State	Weight

As a further step after setting the taxonomy, questions are classified based on that taxonomy using two main approaches: rule-based classifiers and machine learning classifiers.

Apparently, the rule-based classifier is a straightforward way to classify a question according to a taxonomy using a set of predefined heuristic rules. The rules could be just simple as, for example, the questions starting with "Where" are classified as of type LOCATION, etc. Many researchers adopted this approach due to its easiness and quickness such as Moldovan *et al.* [8], Hermjakob [18], as well as Radev *et al.* [15] who used both approaches, the rule-based and machine learning classifiers.

In machine learning approach, a machine learning model is designed and trained on an annotated corpus composed of labeled questions. The approach assumes that useful patterns for later classification will be automatically captured from the corpus. Therefore, in this approach, the choice of features (for representing questions) and classifiers (for automatically classifying questions into one or several classes of the taxonomy) are very important. Features may vary from simple surface of word or morphological ones to detailed syntactic and semantic features using linguistics analysis. Hermjakob [18] used machine learning based parsing and question classification for question-answering. Zhang and Lee [19] compared various choices for machine learning classifiers using the hierarchical taxonomy proposed by Li and Roth [17], such as: Support Vector Machines (SVM), Nearest Neighbors (NN), Naïve Bayes (NB), Decision Trees (DT), and Sparse Network of Winnows (SNoW).

- **Information Retrieval:**

Stoyanchev *et al.* [6] presented a document retrieval experiment on a question answering system, and evaluated the use of named entities and of noun, verb, and prepositional phrases as exact match phrases in a document retrieval query. Gaizauskas and Humphreys [20] described an approach to question answering that was based on linking an IR system with an NLP system that performed reasonably thorough linguistic analysis. While Kangavari *et al.* [21] presented a simple approach to improve the accuracy of a question answering system using a knowledge database to directly return the same answer for a question that was previously submitted to the QA system, and whose answer has been previously validated by the user.

- **Answer Extraction:**

Ravichandran and Hovy [22] presented a model for finding answers by exploiting surface text information using manually constructed surface patterns. In order to enhance the poor recall of the manual hand-crafting patterns, many researchers acquired text patterns automatically such as Xu *et al.* [23]. Also, Peng *et al.* [24] presented an approach to capture long-distance dependencies by using linguistic structures to enhance patterns. Instead of exploiting surface text information using patterns, many other researchers such as Lee *et al.* [25] employed the named-entity approach to find an answer.

Tables (4) and (5) show a comparative summary between the aforementioned researches with respect to the QA components and the QA approaches, respectively. (Table 4) illustrates the different QA system components that were covered by each of the aforementioned researches, while (Table 5) shows the approaches that were utilized by each research within every component.

Table 4: The QA components covered by QA research

	Question Processing			Document Processing			Answer Processing		
QA Research \ QA Components	Question Analysis	Question Classification	Question Reformulation	Information Retrieval	Paragraph Filtering	Paragraph Ordering	Answer Identification	Answer Extraction	Answer Validation
Gaizauskas & Humphreys (QA-LaSIE) [20]		✓		✓			✓	✓	
Harabagiu *et al.* (FALCON) [16]		✓	✓	✓	✓	✓	✓	✓	✓
Hermjakob *et al.* [18]		✓					✓		
Kangavari *et al.* [21]	✓	✓	✓	✓	✓	✓	✓	✓	✓
Lee *et al.* (ASQA) [25]	✓	✓		✓			✓	✓	
Li & Roth [17]		✓					✓		
Moldovan *et al.* (LASSO) [8]	✓	✓	✓	✓	✓	✓	✓	✓	
Peng *et al.* [24]		✓		✓			✓	✓	
Radev *et al.* (NSIR) [15]		✓		✓	✓		✓	✓	
Ravichandran & Hovy [22]		✓		✓			✓	✓	
Stoyanchev *et al.* (StoQA) [6]	✓	✓	✓	✓			✓	✓	
Xu *et al.* [23]		✓		✓			✓	✓	
Zhang & Lee [19]		✓		✓			✓	✓	

Table 5: The QA approaches exploited by QA research

	Question Classification				Information Retrieval		Answer Extraction	
QA Research \ QA Approaches	Flat Taxonomy	Hierarchical Taxonomy	Rule-based Classifier	Machine Learning	Web Corpus	Knowledge-base Corpus	Text Patterns	Named Entity
Gaizauskas & Humphreys (QA-LaSIE) [20]	✓		✓		✓			✓
Harabagiu *et al.* (FALCON) [16]		✓	✓		✓			✓
Hermjakob *et al.* [18]		✓		✓				✓
Kangavari *et al.* [21]		✓	✓		✓	✓	✓	
Lee *et al.* (ASQA) [25]		✓	✓	✓	✓			✓
Li & Roth [17]		✓		✓				
Moldovan *et al.* (LASSO) [8]		✓	✓		✓			✓
Peng *et al.* [24]			✓	✓	✓		✓	✓
Radev *et al.* (NSIR) [15]	✓		✓	✓	✓		✓	
Ravichandran & Hovy [22]							✓	
Stoyanchev *et al.* (StoQA) [6]					✓	✓	✓	✓
Xu *et al.* [23]	✓		✓		✓		✓	
Zhang & Lee [19]		✓		✓				✓

4. Analysis & Discussion

This section discusses and analyzes the aforementioned models proposed by QA researchers in section (3). Researches are presented and discussed in a chronological order describing the main contributions, experimental results, and main limitations for each research. However, as an introductory subsection, the metrics used in evaluating QA systems are first presented to give a thorough explanation and understanding of the meaning behind the experimental results obtained by the QA researches. At the end of the discussion, a following subsection summarizes and concludes what had been analyzed and discussed.

4.1 Evaluation Metrics

The evaluation of QA systems is determined according to the criteria for judging an answer. The following list captures some possible criteria for answer evaluation [1]:

(1) Relevance: the answer should be a response to the question.
(2) Correctness: the answer should be factually correct.
(3) Conciseness: the answer should not contain extraneous or irrelevant information.
(4) Completeness: the answer should be complete (not a part of the answer).
(5) Justification: the answer should be supplied with sufficient context to allow a user to determine why this was chosen as an answer to the question.

Based on the aforementioned criteria, there are three different judgments for an answer extracted from a document:

- "Correct": if the answer is responsive to a question in a correct way - (criteria 1 & 2).
- "Inexact": if some data is missing from or added to the answer - (criteria 3 & 4)
- "Unsupported": if the answer is not supported via other documents - (criterion 5).

The main challenge of most QA systems is to retrieve chunks of 50-bytes, called "short answers" or 250-bytes which are called "long answers", as a requirement of TREC QA Track [10]. However, in order to provide automated evaluation for these answers, each question has pairs of "answers patterns" and "supporting documents identifiers". Therefore, there are two main types of evaluation, namely "lenient" and "strict" evaluations. "Lenient" evaluation uses only the answers patterns without using the supporting documents identifiers, and hence it does not ensure that the document has stated the answer. "Strict" evaluation, on the other hand, uses both the answers patterns along with the supporting documents identifiers.

There are several evaluation metrics that differ from one QA campaign to another (e.g. TREC, CLEF, NTCIR, etc). Moreover, some researchers develop and utilize their own customized metrics. However, the following measures are the most commonly used measures that are typically utilized for automated evaluation:

- **Precision, Recall and F-measure:**

Precision and recall are the traditional measures that have been long used in information retrieval while the F-measure is the harmonic mean of the precision and recall; these three metrics are given by:

$$\text{Precision} = \frac{\text{number of correct answers}}{\text{number of questions answered}}$$

$$\text{Recall} = \frac{\text{number of correct answers}}{\text{number of questions to be answered}}$$

$$\text{F-measure} = \frac{2\,(\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

- **Mean Reciprocal Rank (MRR):**

The Mean Reciprocal Rank (MRR), which was first used for TREC8, is used to calculate the answer rank (relevance):

$$\text{MRR} = \sum_{i=1}^{n} \frac{1}{r_i}$$

where n is the number of test questions and r_i is the rank of the first correct answer for the i-th test question.

- **Confidence Weighted Score (CWS):**

The confidence about the correctness of an answer is evaluated using another metric called Confidence Weighted Score (CWS), which was defined for TREC11:

$$\text{CWS} = \sum_{i=1}^{n} \frac{p_i}{n}$$

where n is the number of test questions and p_i is the precision of the answers at positions from 1 to i in the ordered list of answers.

4.2 Proposed Research – Contributions, Experiments and Limitations

- **Moldovan *et al.* (LASSO)** [8]**, 1999:**

Contribution

Their research relied on NLP techniques in novel ways to find answers in large collections of documents. The question was processed by combining syntactic information with semantic information that characterized the question (e.g. question type or question focus), in which eight heuristic rules were defined to extract the keywords used for identifying the answer. The research also introduced paragraph indexing where retrieved documents were first filtered into paragraphs and then ordered.

Experimental environment and results

The experimental environment was composed of 200 questions of the TREC8 corpus, in which all questions were classified according to a hierarchy of Q-subclasses.

Table 6: Experimental Results – Moldovan *et al.*

	Answers in top 5	MRR score (strict)
Short answer (50-bytes)	68.1%	55.5%
Long answer (250-bytes)	77.7%	64.5%

Limitations

The question was considered to be answered correctly just if it was among the top five ranked long answers. Although, this

was not considered a problem at that time, but starting from TREC-2002, it was required for all QA systems to provide only one answer.

- **Harabagiu *et al.* (FALCON)** [16]**, 2000:**

Contribution

The same developers of LASSO [8] continued their work and proposed another QA system called FALCON which adapted the same architecture of LASSO. The newly proposed system, FALCON, was characterized by additional features and components. They generated a retrieval model for boosting knowledge in the answer engine by using WordNet for semantic processing of questions. Also, in order to overcome the main limitation that appeared in LASSO, they provided a justification option to rule-out erroneous answers to provide only one answer.

Experimental environment and results

The experiments were held on the TREC9 corpus in which questions and document collection were larger than that of TREC8 and of a higher degree of difficulty. The experimental results of FALCON outperformed those of LASSO, which proved that the added features had enhanced the preceding model.

Table 7: Experimental Results – Harabagiu *et al.*

	MRR score (lenient)	MRR score (strict)
Short answer (50-bytes)	59.9%	58.0%
Long answer (250-bytes)	77.8%	76.0%

- **Gaizauskas and Humphreys (QA-LaSIE)** [20]**, 2000:**

Contribution

The research presented an approach based on linking an IR system with an NLP system that performed linguistic analysis. The IR system treated the question as a query and returned a set of ranked documents or passages. The NLP system parsed the questions and analyzed the returned documents or passages yielding a semantic representation for each. A privileged query variable within the semantic representation of the question was instantiated against the semantic representation of the analyzed documents to discover the answer.

Experimental environment and results

Their proposed approach had been evaluated in the TREC8 QA Track. They tested the system with two different IR engines under different environments, in which the best achieved results were as follows:

Table 8: Experimental Results – Gaizauskus & Humphreys

	Precision	Recall
Short answers (50-bytes)	26.67%	16.67%
Long answers (250-bytes)	53.33%	33.33%

Limitations

The overall success of the approach was limited, as only two-thirds of the test set questions was parsed. Also, the QA-LaSIE system employed a small number of business domain ontology although the QA system was intended to be general (open-domain).

- **Hermjakob** [18]**, 2001:**

Contribution

The research showed that parsing improved dramatically when the Penn Treebank training corpus was enriched with an additional Questions Treebank, in which the parse trees were semantically enriched to facilitate question-answering matching. The research also described the hierarchical structure of different answer types "Qtargets" in which questions were classified.

Experimental environment and results

In the first two test runs, the system was trained on 2000 and 3000 Wall Street Journal WSJ sentences (enriched Penn Treebank). In the third and fourth runs, the parser was trained with the same WSJ sentences augmented by 38 treebanked pre-TREC8 questions. For the fifth run, 200 TREC8 questions were added as training sentences testing TREC9 sentences. In the final run, the TREC8 and TREC9 questions were divided into five subsets of about 179 questions. The system was trained on 2000 WSJ sentences plus 975 questions.

Table 9: Experimental Results – Hermjakob

No. of Penn sentences	No. of added Q. sentences	Labeled Precision	Labeled Recall	Tagging accuracy	Qtarget acc. (strict)	Qtarget acc. (lenient)
2000	0	83.47%	82.49%	94.65%	63.0%	65.5%
3000	0	84.74%	84.16%	94.51%	65.3%	67.4%
2000	38	91.20%	89.37%	97.63%	85.9%	87.2%
3000	38	91.52%	90.09%	97.29%	86.4%	87.8%
2000	238	94.16%	93.39%	98.46%	91.9%	93.1%
2000	975	95.71%	95.45%	98.83%	96.1%	97.3%

- **Radev *et al.* (NSIR)** [15]**, 2002:**

Contribution

They presented a probabilistic method for Web-based Natural Language Question Answering, called probabilistic phrase reranking (PPR). Their NSIR web-based system utilized a flat taxonomy of 17 classes, in which two methods were used to classify the questions; the machine learning approach using a decision tree classifier, and a heuristic rule-based approach.

Experimental environment and results

The system was evaluated upon the 200 question from TREC8, in which it achieved a total reciprocal document rank of 0.20. The accuracy in classifying questions had been greatly improved using heuristics. Using machine learning, the training error rate was around 20% and the test error rate reached 30%. While the training error in the heuristic approach never exceeded 8% and the testing error was around 18%.

Limitations

The PPR approach did not achieve the expected promising results due to simple sentence segmentation and POS (parts-of-speech) tagging and text chunking. Also, their QA system did not reformulate the query submitted by the user.

- **Ravichandran & Hovy** [22]**, 2002:**

Contribution

They presented a method that learns patterns from online data using some seed questions and answer anchors, without needing human annotation.

Experimental environment and results

Using the TREC10 question set, two set of experiments were performed. In the first one, the TREC corpus was used as the input source using an IR component of their QA system. In the second experiment, the web was used as the input source using AltaVista search engine to perform IR.

Table 10: Experimental Results – Ravichandran & Hovy

Question Type	No. of questions	MRR on TREC docs	MRR on the web
BIRTHYEAR	8	48%	69%
INVENTOR	6	17%	58%
DISCOVERER	4	13%	88%
DEFINITION	102	34%	39%
WHY-FAMOUS	3	33%	0%
LOCATION	16	75%	86%

Limitations

It only worked for certain types of questions that had fixed anchors, such as "where was X born". Therefore, it performed badly with general definitional questions, since the patterns did not handle long-distance dependencies.

- **Li & Roth** [17]**, 2002:**

Contribution

Their main contribution was proposing a hierarchical taxonomy in which questions were classified and answers were identified upon that taxonomy. Li and Roth used and tested a machine learning technique called SNoW in order to classify the questions into coarse and fine classes of the taxonomy. They also showed through another experiment the differences between a hierarchical and flat classification of a question.

Experimental environment and results

Their experiments used about 5500 questions divided into five different sizes datasets (1000, 2000, 3000, 4000, 5500), collected from four different sources. These datasets were used to train their classifier, which was then tested using 500 other questions collected from TREC10. Their experimental results proved that the question classification problem can be solved quite accurately using a learning approach.

Limitations

The research did not consider or test other machine learning classifiers that could have achieved more accurate results than SNoW, and at the same time it did not provide any reason for choosing SNoW in particular over other machine learning algorithms.

- **Zhang and Lee** [19]**, 2003:**

Contribution

This research worked on the limitation of the aforementioned research [17], and carried out a comparison between five different algorithms of machine learning which were: Support Vector Machine (SVM), Nearest Neighbors (NN), Naïve Bayes (NB), Decision Tree (DT) and Sparse Network of Winnows (SNoW). Furthermore, they proposed a special kernel function called tree kernel that was computed efficiently by dynamic programming to enable the SVM to take advantage of the syntactic structures of questions which were helpful to question classification.

Experimental environment and results

Under the same experimental environment used by Li and Roth [17], all learning algorithms were trained on five different sizes training datasets and were then tested on TREC10 questions. The experimental results proved that the SVM algorithm outperformed the four other methods in classifying questions either under the coarse-grained category (Table 11), or under the fine-grained category (Table 12). The question classification performance was measured by accuracy, i.e. the proportion of correctly classified questions among all test questions.

Table 11: Experimental Results (coarse-grained) – Zhang & Lee

Algorithm	1000	2000	3000	4000	5500
NN	70.0%	73.6%	74.8%	74.8%	75.6%
NB	53.8%	60.4%	74.2%	76.0%	77.4%
DT	78.8%	79.8%	82.0%	83.4%	84.2%
SNoW	71.8%	73.4%	74.2%	78.2%	66.8%
SVM	76.8%	83.4%	87.2%	87.4%	85.8%

Table 12: Experimental Results (fine-grained) – Zhang & Lee

Algorithm	1000	2000	3000	4000	5500
NN	57.4%	62.8%	65.2%	67.2%	68.4%
NB	48.8%	52.8%	56.6%	56.2%	58.4%
DT	67.0%	70.0%	73.6%	75.4%	77.0%
SNoW	42.2%	66.2%	69.0%	66.6%	74.0%
SVM	68.0%	75.0%	77.2%	77.4%	80.2%

- **Xu *et al.*** [23]**, 2003:**

Contribution

For definitional QA, they adopted a hybrid approach that used various complementary components including information retrieval and various linguistic and extraction tools such as name finding, parsing, co-reference resolution, proposition extraction, relation extraction and extraction of structured patterns.

Experimental environment and results

They performed three runs using the F-metric for evaluation. In the first run, BBN2003A, the web was not used in answer finding. In the second run, BBN2003B, answers for factoid questions were found using both TREC corpus and the web while list questions were found using BBN2003A. Finally, BBN2003C was the same as BBN2003B except that if the answer for a factoid question was found multiple times in the corpus, its score was boosted.

Table 13: Experimental Results (Definitional QA) – Xu *et al.*

BBN2003A	BBN2003B	BBN2003C	Baseline
52.1%	52.0%	55.5%	49.0%

Limitations
The experiments tested only the "what" and "who" questions, while other factoid questions such as "when" and "where" were not experimented.

- **Peng *et al.*** [24]**, 2005:**

Contribution
Their research presented an approach to handle the main limitations of Ravichandran & Hovy [22]. They explored a hybrid approach for Chinese definitional question answering by combining deep linguistic analysis (e.g. parsing, co-reference, named-entity) and surface pattern learning in order to capture long-distance dependencies in definitional questions.
Experimental environment and results
They produced a list of questions and identified answer snippets from TDT4 data. The overall results showed that combining both pure linguistic analysis and pure pattern-based systems improved the performance of definitional questions, which proved that linguistic analysis and pattern learning were complementary to each other, and both were helpful for definitional questions.
Limitations
The pattern matching was based on simple POS tagging which captured only limited syntactic information without providing any semantic information.

- **Lee *et al.* (ASQA)** [25]**, 2005:**

Contribution
Their research proposed hybrid architecture for the NTCIR5 CLQA to answer Chinese factoid questions. They presented a knowledge-based approach (InfoMap) and a machine learning approach (SVM) for classifying Chinese questions. They adopted and integrated several approaches from preceding research, such as question focus [8], coarse-fine taxonomy [17], and SVM machine learning [19].
Experimental environment and results
In the CLQA C-C task (Cross-Language Question-Answering from Chinese-to-Chinese) of NTCIR campaign, their system achieved overall accuracy of 37.5% for correct answers and 44.5% for correct unsupported answers. Also, the accuracy of their question classification module was 88% using InfoMap and 73.5% using SVM.

- **Stoyanchev *et al.* (StoQA)** [6]**, 2008:**

Contribution
In their research, they presented a document retrieval experiment on a question answering system. They used exact phrases, which were automatically identified from questions, as constituents to search queries. The process of extracting phrases was performed with the aid of named-entity (NE) recognition, stop-word lists, and parts-of-speech taggers.
Experimental environment and results
The QA system was evaluated using two datasets: the AQUAINT corpus, a 3GB collection of news documents used in TREC-2006; and the other dataset was the web. The datasets used 387 TREC questions with non-empty answers. The documents in the AQUAINT corpus were indexed using Lucene engine. Their experiments utilized automatically and manually created phrases. The automatically created phrases were obtained by extracting nouns, verbs, and propositional phrases, while the manually created phrases were obtained by hand-correcting these automatic annotations.

Table 14: Experimental Results – Stoyanchev *et al.*

	MRR	Overall Recall	Precision of first answer
IR with Lucene on AQUAINT corpus			
Baseline (words disjunction from target & question)	31.4%	62.7%	22.3%
Baseline (+ auto phrases)	33.2%	65.3%	23.6%
Words (+ auto NEs & phrases)	31.6%	65.3%	22.0%
Baseline (+ manual phrases)	29.1%	60.9%	19.9%
Words (+ manual NEs & phrases)	29.4%	60.9%	20.2%
IR with Yahoo API on WEB corpus			
Baseline (words disjunction)	18.3%	57.0%	10.1%
Cascaded (using auto phrases)	22.0%	60.4%	14.0%
Cascaded (using manual phrases)	24.1%	61.4%	15.5%

Limitations
The experimental results showed that the overall accuracy on the web was lower than that on the AQUAINT corpus.

- **Kangavari *et al.*** [21]**, 2008:**

Contribution
The research presented a model for improving QA systems by query reformulation and answer validation. The model depends on previously asked questions along with the user feedback (voting) to reformulate questions and validate answers through the domain knowledge database.
Experimental environment and results
The system worked on a closed aerologic domain for forecasting weather information. Results showed that, from a total of 50 asked questions, the model achieved an improvement of 92%.
Limitations
The model was tested in a restricted experimental environment in which the domain was very specific and the number of questions is relatively small. Also, depending only on the users as a single source for validating answers is a two-edged weapon.

4.3 Conclusion of Analysis & Discussion

The period within (1999-2007) was very rich in QA research than any other time. This was most likely because of the challenging research environment provided by QA tracks of the TREC annual conferences within that period. However, other campaigns such as CLEF and NTCIR still represent a vital vein for QA research.
Also, most researches in the QA field were somehow heterogeneous with respect to their system architecture, approaches, scope, evaluation metrics, etc. But, on the other

hand researches were mainly concerned with one or more of the three basic components of QA systems: question classification, document retrieval and answer extraction, which combine techniques from natural language processing (NLP), information retrieval (IR), and information extraction (IE), respectively.

5. Survey Conclusion

This survey paper, like any other survey, provides a service to the scientific community. It summarized and organized recent research results in a novel way that integrated and added understanding to work in the question-answering field. It emphasized the classification of the existing literature, developing a perspective on the area, and evaluating trends. However, because it is impossible for a survey to include all or even most of previous research, this survey included only the work of the top-publishing and top-cited authors in the QA field. Moreover, because scientific research is a progressive, continuous and accumulative activity, this survey also included research containing minor limitations to show how these limitations were discovered, faced and treated by other researchers.

6. References

[1] L. Hirschman and R. Gaizauskas, "Natural language question answering: the view from here," *Natural Language Engineering,* vol. 7, no. 4, pp. 275-300, 2001.

[2] D. Zhang and W. Lee, "A Web-based Question Answering System," Massachusetts Institute of Technology (DSpace@MIT), 2003.

[3] P. Banerjee and H. Han, "Drexel at TREC 2007: Question Answering," in *Proceedings of the Sixteenth Text Retrieval Conference (TREC 2007)*, 2007.

[4] M. M. Sakre, M. M. Kouta and A. M. N. Allam, "Automated Construction of Arabic-English Parallel Corpus," *Journal of the Advances in Computer Science,* vol. 3, 2009.

[5] M. Ramprasath and S. Hariharan, "A Survey on Question Answering System," *International Journal of Research and Reviews in Information Sciences (IJRRIS),* pp. 171-179, 2012.

[6] S. Stoyanchev, Y. Song and W. Lahti, "Exact phrases in information retrieval for question answering," in *Proceedings of the 2nd workshop on Information Retrieval for Question Answering*, 2008.

[7] A. Lampert, "A Quick Introduction to Question Answering," *CSIRO ICT Centre,* 2004.

[8] D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Goodrum, R. Girju and V. Rus, "Lasso: A Tool for Surfing the Answer Net," in *Proceedings of the Eighth Text Retrieval Conference (TREC-8)*, 1999.

[9] M. M. Sakre, M. M. Kouta and A. M. N. Allam, "Weighting Query Terms Using Wordnet Ontology," *International Journal of Computer Science and Network Security,* vol. 9, no. 4, pp. 349-358, 2009.

[10] E. Voorhees, "Overview of the TREC 2002 Question Answering Track," in *Proceedings of the Text Retrieval Conference (TREC 2002)*, 2002.

[11] B. F. Green, A. K. Wolf, C. Chomsky and K. Laughery, "BASEBALL: An automatic question answerer," in *Proceedings of Western Joint IRE-AIEE-ACM Computing Conference*, Los Angeles, 1961.

[12] W. Woods, "Progress in Natural Language Understanding: An Application to Lunar Geology," in *Proceedings of the National Conference of the American Federation of Information Processing Societies*, 1973.

[13] C. Paris, "Towards More Graceful Interaction: A Survey of Question-Answering Programs," Columbia University Computer Science Technical Reports, 1985.

[14] W. G. Lehnert, The Process of Question Answering - A Computer Simulation of Cognition, Yale University, 1977.

[15] D. Radev, W. Fan, H. Qi, H. Wu and A. Grewal, "Probabilistic Question Answering on the Web," *Journal of the American Society for Information Science and Technology,* vol. 56, no. 6, pp. 571-583, 2005.

[16] S. Harabagiu, D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Girju, V. Rus and P. Morarescu, "FALCON: Boosting Knowledge for Answer Engines," in *Proceedings of the Ninth Text Retrieval Conference (TREC9)*, 2000.

[17] X. Li and D. Roth, "Learning question classifiers," in *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, 2002.

[18] U. Hermjakob, "Parsing and Question Classification for Question Answering," in *Proceedings of the Workshop on Open-Domain Question Answering at ACL-2001*, 2001.

[19] D. Zhang and W. Lee, "Question Classification using Support Vector Machines," in *Proceedings of the 26th Annual International ACM SIGIR Conference*, 2003.

[20] R. Gaizauskas and K. Humphreys, "A Combined IR/NLP Approach to Question Answering Against Large Text Collections," in *Proceedings of the 6th Content-based Multimedia Information Access (RIAO-2000)*, 2000.

[21] M. Kangavari, S. Ghandchi and M. Golpour, "Information Retrieval: Improving Question Answering Systems by Query Reformulation and Answer Validation," *World Academy of Science, Engineering and Technology,* pp. 303-310, 2008.

[22] D. Ravichandran and E. Hovy, "Learning Surface Text Patterns for a Question Answering System," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

[23] J. Xu, A. Licuanan and R. Weischedel, "TREC2003 QA at BBN: Answering Definitional Questions," in *Proceedings of the 12th Text Retrieval Conference*, 2003.

[24] F. Peng, R. Weischedel, A. Licuanan and J. Xu, "Combining deep linguistics analysis and surface pattern learning: A

hybrid approach to Chinese definitional question answering," in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language*, 2005.

[25] C. Lee, C. Shih, M. Day, T. Tsai, T. Jiang, C. Wu, C. Sung, Y. Chen, S. Wu and W. Hsu, "ASQA: Academia Sinica Question Answering System for NTCIR-5 CLQA," in *Proceedings of NTCIR-5 Workshop Meeting*, Tokyo, 2005.