TURKISH FACTOID QUESTION ANSWERING USING ANSWER PATTERN MATCHING

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Nagehan Pala Er

July, 2009

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. İlyas Çiçekli (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Fazlı Can

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Ferda Nur Alpaslan

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute

ABSTRACT

TURKISH FACTOID QUESTION ANSWERING USING ANSWER PATTERN MATCHING

Nagehan Pala Er
M.S. in Computer Engineering
Supervisor: Asst. Prof. Dr. İlyas Çiçekli
July, 2009

Efficiently locating information on the Web has become one of the most important challenges in the last decade. The Web Search Engines have been used to locate the documents containing the required information. However, in many situations a user wants a particular piece of information rather than a document set. Question Answering (QA) systems have addressed this problem and they return explicit answers to questions rather than set of documents. Questions addressed by QA systems can be categorized into five categories: factoid, list, definition, complex, and speculative questions. A factoid question has exactly one correct answer, and the answer is mostly a named entity like person, date, or location. In this thesis, we develop a pattern matching approach for a Turkish Factoid QA system. In TREC-10 QA track, most of the question answering systems used sophisticated linguistic tools. However, the best performing system at the track used only an extensive list of surface patterns; therefore, we decided to investigate the potential of answer pattern matching approach for our Turkish Factoid QA system. We try different methods for answer pattern extraction such as stemming and named entity tagging. We also investigate query expansion by using answer patterns. Several experiments have been performed to evaluate the performance of the system. Compared with the results of the other factoid QA systems, our methods have achieved good results. The results of the experiments show that named entity tagging improves the performance of the system.

Keywords: Factoid question answering, pattern matching, query expansion.

ÖZET

YANIT ÖRÜNTÜSÜ EŞLEŞTİRME YÖNTEMİ İLE TÜRKÇE TEKİL YANITLI SORU YANITLAMA

Nagehan Pala Er
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Yrd. Doç. Dr. İlyas Çiçekli
Temmuz, 2009

Aranan bilgiyi Web'de etkili bir şekilde bulmak, son on yıldaki en zorlu problemlerden biri olmuştur. Aranan bilgiyi içeren belgelerin bulunması için Web Arama Motorları kullanılmaktadır. Ancak, bir çok durumda kullanıcı bir belge kümesinden çok belirli bir bilgiye ihtiyaç duyar. Soru Yanıtlama sistemleri bu problemi adreslemektedir. Soru yanıtlama sistemleri bir sorunun yanıtı olarak bir belge kümesi yerine açık yanıtlar döndürürler. Soru yanıtlama sistemlerinin yanıtladığı sorular beş sınıfa ayrılabilir: tekil yanıtlı, liste, tanım, karmaşık, ve kurgusal sorular. Tekil yanıtlı bir sorunun tam olarak tek bir yanıtı vardır ve bu yanıt genellikle kişi, tarih ve yer gibi bir varlık ismidir. Bu tez kapsamında, Türkçe Tekil Yanıtlı Soru Yanıtlama için örüntü eşleştirme yaklaşımı geliştirdik. TREC-10 Soru Yanıtlama kulvarında yarışan soru yanıtlama sistemlerinden birçoğu gelişmiş dilbilimsel araçlar kullanmıştır. Ancak, bu kulvardaki en başarılı soru yanıtlama sistemi sadece çok miktarda yüzeysel örüntü kullanmıştır. Bu nedenle, biz de Türkçe Tekil Yanıtlı Soru Yanıtlama için yanıt örüntüsü eşleştirme yaklaşımının potansiyelini araştırmaya karar verdik. Yanıt örüntüsü çıkarmak için gövdeleme ve varlık isimleri işaretleme içeren yöntemler denedik. Yanıt örüntülerini sorgu genişletme için de kullandık. Sistemin performansını değerlendirmek için bir çok deney yaptık. Diğer tekil yanıtlı soru yanıtlama sistemlerinin performansları ile karşılaştırıldğında, yöntemlerimiz iyi sonuçlar vermektedir. Yapılan deneyler, varlık isimleri işaretleme yönteminin sistemin performansını artırdığını göstermektedir.

Anahtar sözcükler: Tekil yanıtlı soru yanıtlama, örüntü eşleştirme, sorgu genişletme.

Acknowledgement

I would like to express my gratitude to Asst. Prof. Dr. İlyas Çiçekli, from whom I have learned a lot, due to his supervision, suggestions, and support during this research.

I am also indebted to Prof. Dr. Fazlı Can and Assoc. Prof. Dr. Ferda Nur Alpaslan for showing keen interest to the subject matter and accepting to read and review this thesis.

I am grateful to Bilkent University for providing me scholarship for my MSc study.

I acknowledge the Scientific and Technical Research Council of Turkey (TÜBİTAK) for supporting my MSc studies under MSc Fellowship Program.

I am thankful to my company ASELSAN Inc. for letting and supporting my thesis.

I am very grateful to my mother, my father, my sister, and my brother for giving me encouragement during this thesis and all kind of supports during my life.

I want to thank my husband, Ersin Er, for his patience and help. This thesis would have been impossible without his encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Question Answering

There is a large amount of textual data on a variety of digital mediums such as digital archives, the Web and the hard drives of our personal computers. Efficiently locating information on these digital mediums has become one of the most important challenges in the last decade.

Search engines have been used to locate the documents which are related to user information need. Natural language questions are the best way of expressing user information need but these questions cannot be used directly by search engines. A natural language question is transformed into a query which is a set of keywords. These keywords describe the user information need. After a query is entered into a search engine, the search engine retrieves a set of documents that are ranked according to their relevance to the query. This task is encompassed in Information Retrieval field [2]. To find the desired information, the user reads through the returned document set. However, in many situations a user wants a particular piece of information rather than a document set. Question Answering (QA) which is a kind of Information Retrieval has addressed this problem. The benefit of Question Answering Systems is two-fold: (1) they take natural language questions rather than queries, (2) they return explicit answers rather than set of

documents.

Question Answering is the task of returning a particular piece of information in response to a natural language question. The aim of a question answering system is to present the needed information directly, instead of documents containing potentially relevant information.

Question	Question Type
(1) "Türkiye'nin başkenti neresidir?"	Factoid Question
(2) "Dolmabahçe Sarayı nerededir?"	Factoid Question
(3) "Puslu Kıtalar Atlası kitabının yazarı kimdir?"	Factoid Question
(4) "Barış Manço'nun doğum tarihi nedir?"	Factoid Question
(5) "Eşkiya filminde rol alan oyuncular kimlerdir?"	List Question
(6) "Asya kıtasında hangi ülkeler bulunmaktadır?"	List Question
(7) "Cahit Arf kimdir?"	Definition Question
(8) "Karasal iklim nedir?"	Definition Question
(9) "Avusturya'nın başkentinin nüfusu nedir?"	Complex Question
(10) "Merkez Bankası faizleri düşürecek mi?"	Speculative Question
(11) "Otomobil Endüstrisi kötü durumda mı?"	Speculative Question

Table 1.1: Some questions and their question types

Questions can be divided into five categories regarding the input of question answering systems [14]: factoid questions, list questions, definition questions, complex questions, and speculative questions. Table 1.1 shows some natural language questions in Turkish along with their question types.

A **factoid question** has exactly one correct answer which can be extracted from short text segments. Question Answering systems which deal with factoid questions are called Factoid Question Answering systems. The difficulty level of factoid questions is lower than the other categories. Factoid Question Answering is the main topic of this thesis, and it is detailed in the following section. Questions (1), (2), (3) and (4) in Table 1.1 are examples of factoid questions. For instance, the answer of question (1) is "Ankara" and it can be extracted from the following passages.

> Görüşme süreci içinde AB adayı Türkiye'nin başkenti Ankara için yapılabilecek, yapılması gerekli pek çok şey var. ...
> Ankara, Türkiye Cumhuriyeti Devletinin başkenti ve yönetim merkezidir. ...
> Kitaptaki olaylar, Ankara'nın Türkiye'nin başkenti oluşunun o heyecanlı günlerinde geçiyor. ...

A **list question** expects a list as its answer. Question Answering systems which deal with list questions are called List Question Answering systems. List Question Answering systems assemble a set of distinct and complete exact answers as responses to questions like (5) and (6). For instance, the answers for question (5) can be extracted from the following passages. Each answer phrase is underlined in the passages.

> Başrollerini Şener Şen ve Uğur Yücel'in paylaştığı Eşkiya filmi Türk sineması için bir dönüm noktası olmuştur. ...
> Eşkiya filminde Emel karakterini canladıran Yeşim Salkım, rol arkadaşı Uğur Yücel'e desteği için teşekkür etti. ...
> Özkan Uğur ilk oyunculuk denemelerinden birini Eşkiya filmi ile yaptı....
> Baran'ın (Şener Şen) en yakın arkadaşı olan Berfo (Kamran Usluer), arkadaşına ihanet eder ve Keje (Sermin Hürmeriç) ile evlenir. ...

List QA systems must identify many candidate answers and collect evidence supporting each of the candidate answers to effectively rank them. A common method is interpreting a list question as a factoid question and finding the best answers [19]. Low-ranked answers are removed according to a given threshold. However, factoid answer processing techniques based upon redundancy and frequency counting do not work satisfactorily on list questions, because List QA systems must return all different answers including less-frequent answers. TREC-12 addressed List QA task. The results of TREC-12 [26] show that List QA systems severely suffer from two general problems: *low recall* and *non-distinctive answers*. Since traditional List QA systems operating on large text collections are designed as precision-oriented rather than recall-oriented systems, as the number of expected answers increases, the performance of the systems decreases. Part of

the reason is the use of a document retrieval phase, which limits the number of documents being searched for potential answers, which also limits the number of potential answers.

The answer of a **definition question** is a list of complementary short phrases or sentence fragments from different documents. Questions that ask about the biography of a person such as question (7) or the definition of a thing such as question (8) are categorized as definition question. Answering this type of questions requires more sophisticated methods to piece together relevant text segments extracted from a set of relevant documents.

A **complex question** contains sub-questions so the question is decomposed into sub-questions. Each sub-question can be answered individually and they have to be answered first. Then, the individual responses are combined into an answer that is the answer of original complex question. Syntactic and semantic decomposition strategies are developed to decompose a complex question and they combine natural language processing and reasoning [13]. For example, question (9) is a complex question and it can be decomposed into two factoid questions:

(9.1) "Avusturya'nın başkenti neresidir?"

(9.2) "Viyana'nın nüfusu nedir?"

The original complex question asks the population of the capital of Austria. Firstly, the capital of Austria is identified by the first sub-question (9.1). Then, the answer of the first sub-question is used in the second sub-question (9.2). The answer of the first sub-question is "Viyana" and the second sub-question asks the population of "Viyana". The answer of the second sub-question is also the response for the original complex question.

To answer a **speculative question**, it is necessary to use reasoning techniques and knowledge bases. Question (10) and (11) are examples of speculative questions. Generally, the answer of a speculative question is not explicitly stated in documents so queries are created from the speculative question to collect pieces of the answer. Knowledge bases clustered by the question topic and reasoning

techniques such as temporal reasoning, spatial reasoning, and evidential reasoning are used to piece together the collected information.

In this thesis, we develop a pattern matching approach for Factoid Question Answering. List, definition, complex, and speculative questions are out of the scope of this thesis. At TREC-10 QA track [25], most of the question answering systems used Natural Language Processing (NLP) tools such as parser, WordNet [7], etc. However, the best performing system at TREC-10 QA track used only an extensive list of surface patterns [22]. We therefore decided to investigate their potential for Turkish Factoid Question Answering. We try different methods for answer pattern extraction such as stemming and named entity tagging. We also investigate query expansion by using answer patterns.

1.2 Factoid Question Answering

Factoid Question Answering is the simplest form of question answering. The answers are simple facts; especially these facts are named entities like person, date, or location. Table 1.2 shows some factoid questions in Turkish and their answers.

Question	Answer
"Türkiye'nin başkenti neresidir?"	Ankara
"Dolmabahçe Sarayı neresidir?"	İstanbul
"Puslu Kıtalar Atlası kitabının yazarı kimdir?"	İhsan Oktay Anar
"Barış Manço'nun doğum tarihi nedir?"	2 Ocak 1943

Table 1.2: Factoid questions and their answers

Each of these answers can be found in a short passage that contains the named entity tag of the expected answer. However, the wording of the question and the wording of the passages containing the answer can be different. To solve the mismatch between the question and answer form, both question and candidate answer passages are processed and a similarity measure between the question and candidate answer passages are assigned.

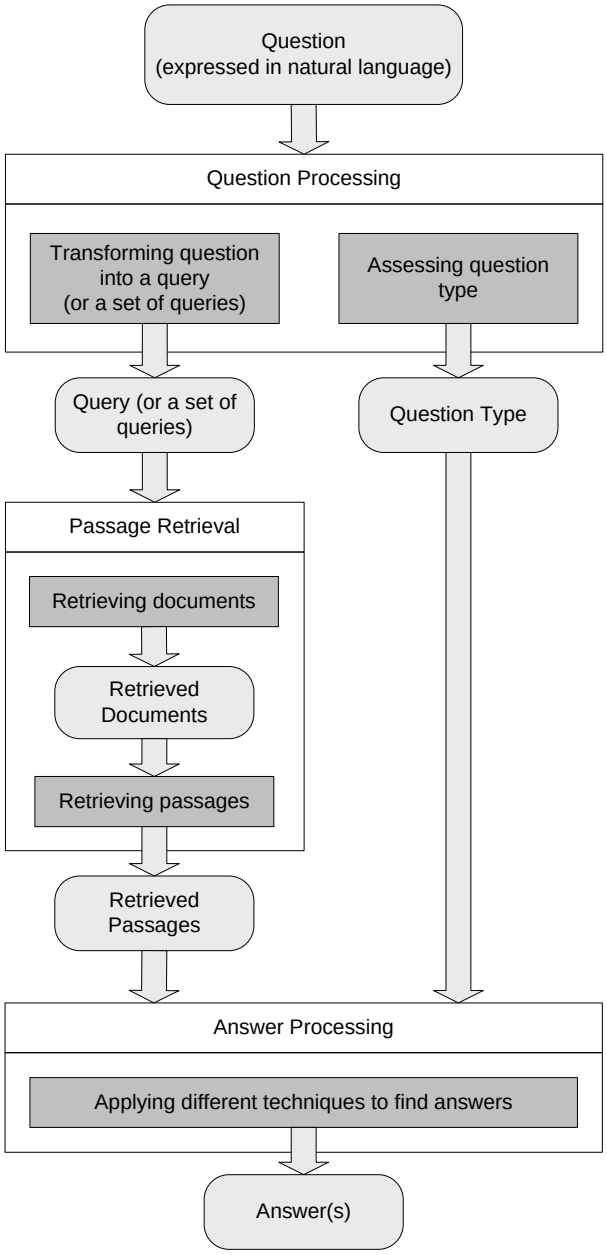


Figure 1.1: Conceptual architecture of a typical Factoid QA System

Figure 1.1 shows a conceptual architecture of a typical Factoid QA System. Many of Factoid Question Answering systems comprise of following three phases [12] and these phases are explained in the following sections:

1. Question Processing

2. Document/Passage Retrieval

3. Answer Processing

1.2.1 Question Processing

Questions are first analyzed in the question processing phase. Two sub-tasks are performed in this phase: (1) transforming the question into a query or queries and (2) assessing the question type.

1.2.1.1 Transforming Question into Query(ies)

The first task in question processing is to transform the natural language question into a query or queries. Different query formation approaches can be applied to transform the natural language question into a query. Basic approach is to form a keyword from each word in the question. Generally, question words (*nerede, ne zaman,* etc.) and stopwords (*ve, bu, defa,* etc.) are removed. Alternatively, keywords can be created from only the words found in the noun phrases in the question. Another approach is to apply query expansion methods which add query terms in order to match different forms of the answer. Morphological variants of keywords or synonyms of keywords can be added as keywords to the query.

1.2.1.2 Assessing Question Type

The second task in question processing is to assess the type of the question. *Question type* is the name of the relation between the question phrase and its answer phrase. Question type associates the question with its *answer type*. Answer type is the Named Entity (NE) Tag of the expected answer.

Question typologies can be coarse-grained or fine-grained. A coarse-grained question typology consists of coarse-grained question types like PERSON, DATE,

CITY, etc. which are direct matches of the answer types. A fine-grained question typology contains fine-grained question types like CAPITAL-OF-COUNTRY, PLACE-OF-BIRTH, DATE-OF-BIRTH, etc. These question types are classified under the associated answer type. For example, CAPITAL-OF-COUNTRY question type is classified under its associated answer type CITY. Webclopedia question typology is an example question typology that was suggested by [10]. Example question types are given in the following list.

- *CAPITAL-OF-COUNTRY* question type defines the relation between a country and the capital of that country.
- *PLACE-OF-BIRTH* question type defines the relation between a person and the place where the person was born.
- *DATE-OF-BIRTH* question type defines the relation between a person and the date which the person was born.
- *ACTOR* question type defines the relation between a person and a film in which the person acted.
- *POPULATION* question type defines the relation between a city/country and the population of that city/country.
- *ABBREVIATION* question type defines the relation between an abbreviation and the meaning which the abbreviation stands for.

Question Patterns can be used to identify question types. Question patterns are regular expressions. A set of question patterns is associated with a question type. If a question matches with one of these question patterns, the question type is assessed as the associated question type of the matched question pattern. Webclopedia question typology [10] includes 276 hand-written question patterns to identify 180 question types. A question pattern example is given below:

"Where was PERSON born"

This question pattern is associated with PLACE-OF-BIRTH question type. If a question matches with this question pattern, its question type is identified as PLACE-OF-BIRTH.

A question type identifier can be built by applying supervised machine learning techniques. These question type identifiers are trained on databases which contain the questions and their hand-assigned question types. Words and named entities in the question can be used as features.

Correct identification of question type is important for correct identification of answer type. Answer types are used by systems as a matching criteria to filter out candidate answers in answer processing, and hence correctness of answers depends on correct identification of question type. If a wrong answer type is assessed, then there is no way to answer correctly the question. Table 1.3 shows the associated answer types of the question types defined above.

Question Type	**Answer Type (NE Tag)**
CAPITAL-OF-COUNTRY	CITY
PLACE-OF-BIRTH	CITY or COUNTRY
DATE-OF-BIRTH	DATE
ACTOR	PERSON
POPULATION	NUMBER
ABBREVIATION	ABBREVIATION

Table 1.3: Some question types and their associated answer types

1.2.2 Document/Passage Retrieval

The techniques used in answer processing such as parsing and named entity tagging are expensive NLP techniques so these techniques cannot be applied on huge amounts of textual data. Information Retrieval methods are applied to get a small number of related documents from huge amounts of textual data.

The first task is called document retrieval. Factoid QA systems use Information Retrieval techniques to retrieve related documents. The query created

in question processing is used to query an Information Retrieval system such as a Web search engine. A set of related documents are returned by document retrieval.

The second task is passage retrieval. Relevant passages are extracted from these related documents. Relevant passages have potential to contain the answer. A basic approach to retrieve passages is to include the keywords used in the query. Another approach is to select passages which contain words whose named entity tag is the same as the named entity tag of the expected answer. Supervised machine learning techniques can be used to combine these different approaches. The following items can be used as features.

- **Number of keywords:** The number of keywords included in the passage
- **Number of keywords in the longest sequence of words:** The number of keywords in the longest exact sequence of words included in the passage
- **Number of named entity words:** The number of words whose named entity tag is the same as the named entity tag of the expected answer
- **Rank of the document:** The rank of the document which contains the passage

Selected passages are passed to answer processing phase. In our system, sentences are retrieved from this phase so the phase is called *Sentence Retrieval*.

1.2.3 Answer Processing

The final phase of Factoid QA is answer processing. A specific answer is extracted from the passages returned by the previous phase. Various techniques have been explored by QA system designers in order to successfully locate the answer. These techniques are explained in the following sections.

1.2.3.1 Answer Type Matching

A named entity tagger is applied to the returned passages and named entity tags of the words in the passages are identified. The passages which do not contain the expected answer type (named entity tag) are filtered out. The words which are tagged with the expected named entity tag are extracted as answer. For example, the answer type of the question "Türkiye'nin başkenti neresidir?" is CITY. The following passage contains a word whose named entity tag is the same with the expected answer type; CITY. Underlined word is extracted as an answer by the answer type matching technique.

> Görüşme süreci içinde AB **(ABBREVIATION)** adayı Türkiye'nin **(COUNTRY)** başkenti Ankara **(CITY)** için yapılabilecek, yapılması gerekli pek çok şey var.

If a passage contains multiple examples of the same named entity tag, all of them are extracted as separate answers. For instance, the following passage contains two words whose named entity tag is CITY. Underlined words are extracted as separate answers.

> Konferansın ilk günü Türkiye'nin **(COUNTRY)** başkenti Ankara'da **(CITY)**, ikinci günü ise Türkiye'nin **(COUNTRY)** en büyük şehri İstanbul'da **(CITY)** gerçekleştirilecek.

The first answer is "Ankara" which is correct answer for our example question and the second answer is "İstanbul" which is an incorrect answer.

1.2.3.2 Answer Pattern Matching

Answer pattern matching technique uses textual patterns to extract answers from the passages returned by passage retrieval. Since the patterns are used in Answer Processing phase, they are called *Answer Patterns*. Answer patterns indicate strings which contain the answer with high probability. Answer patterns are regular expressions and they are matched against the passages for answer extraction. If an answer pattern is matched, the answer is extracted from the passage and

put into the candidate answer list along with the confidence factor of the pattern which has been used to extract it.

Answer patterns can either be written by hand or learned automatically. Whether an answer pattern is written by hand or learned automatically, the answer pattern must have a confidence factor. Confidence factor of an answer pattern is used to assess the reliability of the answer extracted by that answer pattern.

Each question type has its own specific answer patterns. Question type is identified in the question processing phase. Only the answer patterns of the identified question type are used in answer processing phase.

Answer patterns are useful especially when a passage contains multiple examples of the same named entity type. For example, suppose that the question is "Türkiye'nin başkenti neresidir?" and there exists an answer pattern "<Q>'nin başkenti <A>" for CAPITAL-OF-COUNTRY question type. (<Q> stands for question phrase and <A> stands for answer phrase.) Boldfaced part of the passage below matches with the answer pattern and only the underlined word is produced as an answer.

> Konferansın ilk günü **Türkiye'nin başkenti Ankara'da**, ikinci günü ise Türkiye'nin en büyük şehri İstanbul'da gerçekleştirilecek.

The approach described in this thesis is based on Answer Pattern Matching technique. Since writing answer patterns by hand is time consuming and the list of answer patterns is generally far from complete, we learn answer patterns automatically from the Web. A conventional web search engine is used to fetch the documents.

Answer Pattern Matching technique is used by several QA systems such as [16], [17], [22]. It is shown that Answer Pattern Matching is an effective technique to find answers. In this thesis, we extract answer patterns for Turkish by using different answer pattern extraction methods. These methods are compared according to their effectiveness.

We develop an approach for query expansion based on answer patterns. New queries are created from the most reliable answer patterns. The documents returned by these newly created queries have more potential to include answers. The results of query expansion are also discussed.

1.2.3.3 Frequency Counting

After candidate answers are identified by using any method such as answer type matching, answer pattern matching, etc., the candidate answers are sorted according to their frequencies. More frequent answers take precedence over the less frequent answers. The frequency counting technique is based on redundancy, and hence the success rate of the technique increases when it is applied on large text collections such as the Web. Frequency Counting technique relies on correct answers to appear more frequently than other incorrect answers.

The technique can be applied in two ways. When a new candidate answer is added to the list of candidate answers, it is searched in the list and if the same candidate answer is already included in the list,

1. its frequency count is increased by one or
2. its confidence factor is increased by adding the confidence factor of the new candidate answer.

1.2.3.4 Combining Different Techniques

One answer processing technique may not be sufficient to find the correct answer. Combining different answer processing techniques may increase the success of QA systems.

A classifier can be used to combine different answer processing techniques. The information produced from these techniques are used as features of the classifier. The classifier ranks the candidate answers. The features can be as follows:

- **Answer type match:** A boolean feature which is true if the passage contains a phrase whose type is the same as the expected answer type, otherwise false.
- **Answer pattern match:** The identity of the matched answer pattern. An invalid identity is used if there is no match.
- **Number of question keywords:** Number of question keywords which are contained in the passage.

1.3 Related Work

1.3.1 Question Answering

Automating the process of question answering has been studied since the earliest days of computational linguistics. Several QA systems have been developed since the 1960s [20]. The first systems had a targeted domain of expertise so they are called restricted-domain QA systems. An example of such a system is BASEBALL [8] which was able to answer questions about the American baseball league statistics. BASEBALL system used shallow language parsing techniques. Another example system is LUNAR [28] which was designed to answer questions regarding the moon rocks. LUNAR system was one of the first user evaluated question answering systems. In the evaluation, 111 questions were asked to LUNAR system by geologists and %78 of the questions were answered correctly. The similarity between BASEBALL and LUNAR is that they used databases to store their knowledge base. Questions were transformed into database queries. These systems performed well if the questions were inside the targeted domain whereas their performance was poor if the questions were outside the targeted domain. These early QA systems were usually natural language front-ends of highly structured data sources, whereas modern question answering systems aimed to operate on unstructured data.

The first web-based QA systems started to appear around the 1990s. START

[13] system provides answers to natural language questions using knowledge bases mined from the Web. START system analyzes text and produces a knowledge base which annotates the information found in the text. All sentences are annotated as *ternary expressions*, <subject, relation, object>. Ternary expressions are indexed in the knowledge base. In order to answer a question, the question is translated into a ternary expression which is used to search the knowledge base. If the ternary expression matches an entry of the knowledge base, the answer is returned from the matched ternary expression.

FAQ Finder [9] is designed to help users to navigate through already existing FAQ (Frequently Asked Questions) collections. The system organizes FAQ text files into questions, section headings, keywords, etc. and indexes these information. Syntactic parsing is used to identify noun and verb phrases in a question and semantic concept matching is used to select possible matches between the query and target FAQ entries in the index. Semantic concepts are extracted through the use of WordNet [7]. Another automated FAQ answering system is Ask Jeeves [21] which retrieves existing question-answer pairs from its knowledge base. In Ask Jeeves, knowledge base is mined from FAQ collections, and it uses shallow language understanding during matching a user question to FAQ entries in the knowledge base. The matching is based on keyword comparison, and Ask Jeeves does not perform syntactic parsing and does not extract semantic concepts.

AskMSR question answering system [4] depends on data redundancy so the system performs well if a large data resource such as the Web is used. The system first rewrites the question by using hand-built query-to-answer reformulations. For example, *"Where is the Louvre Museum located"* is rewritten as *"The Louvre Museum is located"* or *"The Louvre Museum is in"*. Each query-to-answer reformulation has a confidence factor. The rewritten form of the question is searched in the collection of documents. Returned documents are processed in accordance with the patterns specified by the rewritables. Unigrams, bigrams and trigrams are extracted and their confidence factors are assigned according to the confidence factor of the query-to-answer reformulation which the query is rewritten. These confidence factors are summed across all documents containing the n-gram. These n-grams are filtered out according to expected answer type.

Finally, an answer tiling algorithm is applied to merge similar answers and assembles longer answers from overlapping smaller answer fragments. For example, "A B C" and "B C D" n-grams are merged as "A B C D". AskMSR system does not use sophisticated linguistic analysis of either questions and candidate answers.

Many international question answering contest-type evaluation tasks have been held at conferences and workshops, such as TREC [23], NTCIR [15], and CLEF [5]. The goal of QA tasks is to foster research on question answering systems. TREC QA task was first introduced in 1999. The focus of TREC QA task is to build a fully automatic open-domain question answering system. In the TREC QA task, participants are given a large document set and a set of questions; for each question, the QA system has to return an exact answer to the question and a document which supports that answer. TREC QA task is the major large scale evaluation environment for open-domain QA systems.

Wolfram Alpha [27], a product by the creators of well known Mathematica software, is an online service that answers factoid queries. As it is built on top of a mathematical engine it is suited to answer mathematical questions such as "derivative of x sin x". Wolfram Alpha is also capable of responding to fact-based questions expressed in natural language such as "What is the temperature in Ankara?". There aren't any academic publications about the inner workings of Wolfram Alpha, so we cannot give more information regarding its state with respect to current state of the art in question answering.

1.3.2 Answer Pattern Matching

At the TREC-10 QA track [25], most of the question answering systems used sophisticated linguistic tools, such as parser, named-entity recognizer, WordNet [7], etc. However, the best performing system at the TREC-10 QA track used textual patterns to extract answers [22]. Many question answering system have been stimulated by this result.

The question answering system presented in [22] is based on searching for predefined textual patterns in the candidate answer texts. Each textual pattern has a score which is assigned before question answering. Answer candidates containing the highest-scored textual patterns are chosen as final answers. This technique does not require linguistic or knowledge-based analysis of neither the question nor the answer candidates. The question answering system uses lexical similarity between the question and a candidate answer if no textual pattern is found. Two thirds of correct answers were obtained using textual patterns according to results presented in [22] and this result shows the feasibility of the approach.

The question answering system uses a hand-built library of patterns which are sequences or combinations of string elements, such as letters, digits, punctuation marks, etc. and words/phrases which are accumulated in special lists. For example, posts such as "president", "prime minister", etc. are accumulated in a special list called *list of posts* and titles such as "Dr.", "Mr.", etc. are accumulated in another special list called *list of titles* and they are used in textual patterns. The following patterns are defined to answer questions like "Who is the prime minister of [country name]".

- "[country name]["'s"][term from the list of posts][term from the list of titles][two capitalized words]"
- "[term from the list of posts]["of"][country name][two capitalized words]"

An approach for automatically learning patterns from the Web is presented in [16]. We use a similar approach to learn answer patterns for our question answering system. They developed Webclopedia question typology [10] which includes 180 question types. Hand-written question patterns are used to identify question types. Our question answering system takes question type along with question phrase as input.

Ephyra [18] is an open-domain question answering system and combines different techniques for question processing and answer processing. Ephyra uses

pattern matching approach in both question processing phase and answer processing phase [17]. A set of patterns called *question patterns* is used to interpret questions in question processing phase. A second set of patterns called *answer patterns* is used to extract answers in answer processing phase. Ephyra automatically learns answer patterns using question-answer pairs as training data. When pattern matching approach fails, Ephyra uses backup question processing and answer processing techniques.

Pattern matching approach presented in [29] consists of two parts, fixed pattern matching and partial pattern matching. Fixed pattern matching is similar to our answer pattern matching approach. Partial pattern matching approach is based on the assumption that the answer is usually surrounded by keywords and their synonyms. If a passage contains keywords or their synonyms and a word tagged with the expected answer type, a matching score is assigned to that passage. If the matching score is above a threshold, the word tagged with the expected answer type is extracted as answer.

Answer pattern matching approach is also used by different languages other than English such as Dutch and Turkish. In [11], a question answering system for Dutch questions is described. For a question, zero or more regular expression patterns are generated according to question type. These generated patterns are applied to the entire document collection. Answers are produced by the matched patterns. Unlike our QA system, these regular expression patterns do not have confidence factors, so answer ranking method is based on Frequency Counting. Candidate answers are ranked according to their frequencies which is the number of times each candidate answer string matched.

BayBilmiş [1] is a question answering system for Turkish. Answer pattern matching approach is used to extract answers along with other techniques. BayBilmiş and our system is different in the manner of building pattern libraries. The pattern library of BayBilmiş is hand-built but our pattern library is learned automatically by using question-answer pairs.

1.4 Outline of the Thesis

In the next chapter, we explain our answer pattern matching technique. Learning process of answer patterns is examined in two phases. The first phase is answer pattern extraction which is described in Chapter 3. In Chapter 4, different methods that are used to extract answer patterns are given. Confidence factor assignment is the second phase of the learning process and it is described in Chapter 5. Question answering by answer pattern matching is explained in Chapter 6. Using answer patterns for query expansion and our answer re-ranking approach are explained in Chapter 6. We discuss the evaluation results in Chapter 7. Finally, we conclude the thesis with Chapter 8.

Chapter 2

Answer Pattern Matching Technique

Answer Pattern Matching technique is one of the answer processing techniques defined in Chapter 1. In this chapter, we desrcibe how answer pattern matching technique is realized by our factoid question answering system.

Answer Pattern Matching technique uses *Answer Patterns* to extract answers. An answer pattern defines a relation between *Question Phrase* and its *Answer Phrase*. A general usage of a question phrase and its answer phrase in the same sentence is represented by an answer pattern. Since factoid questions usually ask a property (answer phrase) of a target (question phrase), an answer pattern defines a relation between the target and its property. For instance, the answer patterns of CAPITAL-OF-COUNTRY question type represent the relationship between a country and the capital of that country, the answer patterns of PLACE-OF-BIRTH question type represent the relationship between a person and a place where the person was born, etc.

Answer patterns can either be written by hand or learned automatically. In our system, answer patterns are learned automatically from the Web. Learning phase of answer patterns is explained in Section 2.1.

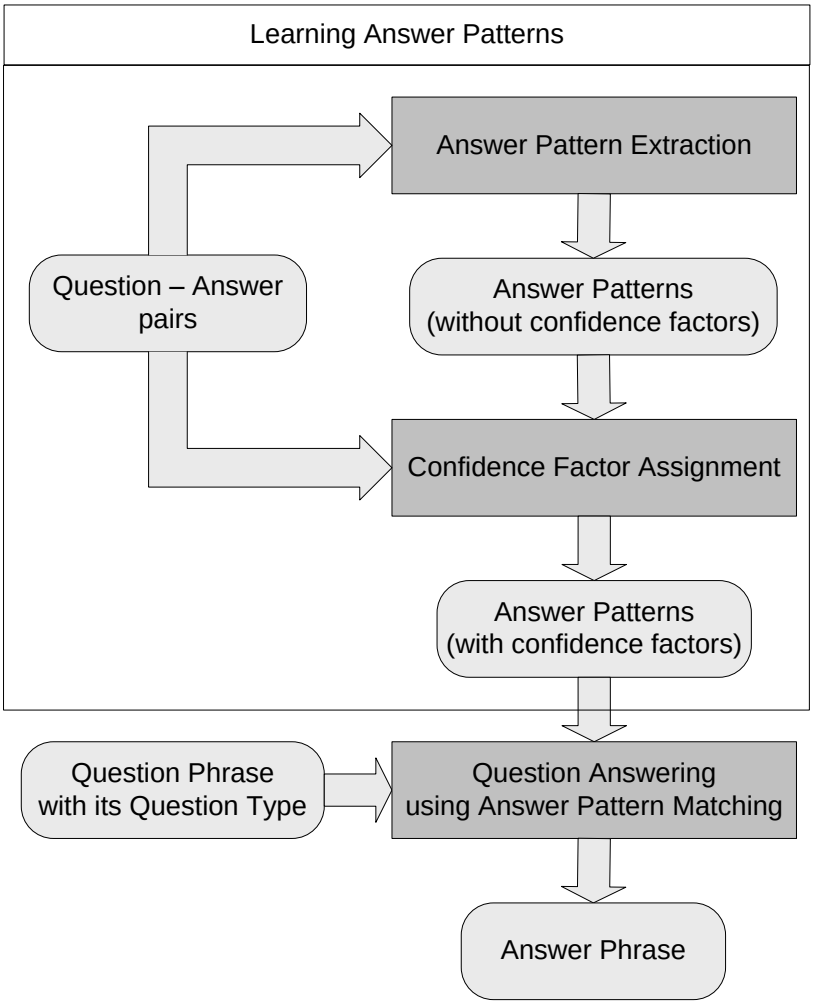


Figure 2.1: Learning and question answering phases and their relationship

After answer patterns are learned for each question type, these patterns are used to extract answers in answer processing phase. Answer patterns are searched in the returned sentences from the sentence retrieval phase. If an answer pattern is found in a passage, an answer is extracted from that passage by the answer pattern. In Section 2.2, question answering using answer pattern matching is described.

Figure 2.1 shows the learning and question answering phases and the relationship between them. After learning phase is completed, a library of answer patterns is built as shown in Figure 2.1. The library of answer patterns is used in the question answering phase.

2.1 Learning Answer Patterns

Answer patterns are used in answer processing phase of our question answering system. The library of answer patterns is built before question answering phase. The library of answer patterns can be hand-built or can be learned. Writing answer patterns by hand is time consuming and the library of answer patterns is usually far from complete. Our question answering system automatically learns answer patterns from the Web. The methods used for relation extraction [6] which is a field in Information Extraction can also be used to learn answer patterns. Since answer patterns represent the relation between the question and its answer, question-answer pairs can be used to extract answer patterns.

Learning answer patterns consists of two phases. In Figure 2.1, first two phases are the phases related with learning answer patterns.

1. Extracting answer patterns
2. Assigning confidence factors to the extracted answer patterns

In the first phase, answer patterns are extracted automatically by using question-answer pairs. For each question type, a set of question-answer pairs is used. Several answer patterns are extracted for each question type. The first phase is explained in Chapter 3 and Chapter 4 in detail.

In the second phase, confidence factors are assigned to the extracted answer patterns by using question-answer pairs. For each question type, the same set of question-answer pairs is used. If extracted answers by an answer pattern are correct, the confidence factor of the answer pattern increases, otherwise, the confidence factor of the answer pattern decreases. The second phase is explained in Chapter 5 in detail.

As shown in Figure 2.1, the same set of question-answer pairs is used in both of the phases. After answer patterns are learned, answer patterns whose confidence factor is under a given threshold are eliminated. The aim of eliminating unreliable answer patters is decreasing the probability of producing incorrect answers.

2.2 Question Answering using Answer Pattern Matching

After answer patterns are learned, the library of answer patterns is used for question answering which is the last phase shown in Figure 2.1. Answer pattern matching approach is applied in answer processing phase of question answering. Question phrase along with its question type is given as input to question answering system. After related sentences are returned from sentence retrieval phase, answer patterns in the library are matched against the sentences for answer extraction. If an answer pattern is matched, the answer is extracted from the passage and put into the candidate answer list along with the confidence factor of the pattern which has been used to extract it. The answers are sorted according to confidence factors. Question answering using answer patterns is explained in Chapter 6.

Our base question answering algorithm creates only a query which includes the question phrase. Since the created query is a general query, the retrieved documents may be insufficient to find the answer. So, we extend our base algorithm to retrieve documents that are more likely to contain answer. Our approach is based on query expansion by using answer patterns which is also described in Chapter 6.

We use an approach to re-rank the list of answers. Our re-ranking approach is based on frequency counting which is described in Chapter 1. After a ranked list of answers are extracted by using answer pattern matching, the list of answers are re-ranked according to their frequencies. More frequent answers take precedence over the less frequent ones. Frequency Counting relies on correct answers to appear more frequently than other incorrect answers. The re-ranking approach is detailed in Chapter 6.

Chapter 3

Answer Pattern Extraction

In this chapter, the first phase of answer pattern learning process is explained. First, an overview of the phase is given and then the steps of the process are explained in detail in the following sections.

3.1 Overview

The basic algorithm that is used to extract answer patterns is as follows:

1. For a question type, prepare a set of question-answer pairs.
2. Query the Web with these pairs and examine the top N returned documents.
3. Break each document into sentences, and keep only sentences containing both the question phrase and answer phrase.
4. Extract a regular expression pattern representing the words and punctuation that occur between and around the two phrases.

Figure 3.1 shows the steps of the answer pattern extraction process. Each step is represented by a rectangle and the input and/or output of a step is represented by a rounded box.

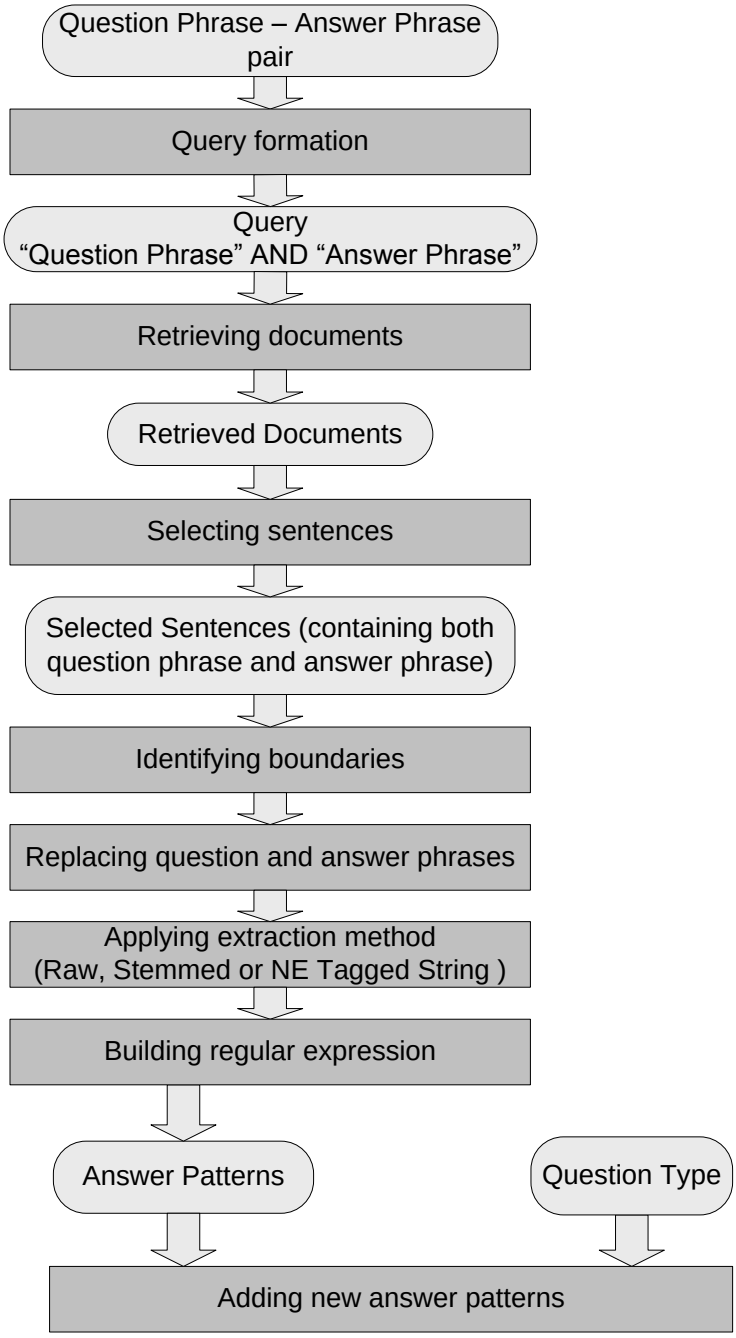


Figure 3.1: Answer pattern extraction process

3.2 Preparing a Set of Question-Answer Pairs

A set of question-answer pairs is prepared for each question type. The set is prepared manually and all pairs have to be correct. As an example, the set used

for CAPITAL-OF-COUNTRY question type is given in Table 3.1. Each line in the table contains a question-answer pair.

Question Phrase	Answer Phrase
türkiye	ankara
fransa	paris
almanya	berlin
bulgaristan	sofya
yunanistan	atina
romanya	bükreş
ingiltere	londra
çin	pekin
rusya	moskova
suriye	şam

Table 3.1: Sample question-answer pairs for answer pattern extraction

The same set of question-answer pairs is used by both phases of the learning process.

3.3 Querying the Web

Each question-answer pair is queried from the Web. Question phrase and answer phrase are AND'ed to form a query. Queries formed for the sample pairs are given in Table 3.2.

We use Bing Web Search Engine [3] to query the Web. Bing Web Search Engine provides a web service for web search. We integrate the web service into our system. The Web search engine retrieves a ranked list of web pages as response to a query. Although the retrieved web pages contain both question phrase and answer phrase, they may not appear in the same sentence.

For each retrieved document, web search engine also returns a snippet which is the summary of the document. Some systems use only the snippets of the returned documents. We use the content of the retrieved documents which requires an additional work of downloading web pages.

Question Phrase	Answer Phrase	Query
türkiye	ankara	“türkiye” AND “ankara”
fransa	paris	“fransa” AND “paris”
almanya	berlin	“almanya” AND “berlin”
bulgaristan	sofya	“bulgaristan” AND “sofya”
yunanistan	atina	“yunanistan” AND “atina”
romanya	bükreş	“romanya” AND “bükreş”
ingiltere	londra	“ingiltere” AND “londra”
çin	pekin	“çin” AND “pekin”
rusya	moskova	“rusya” AND “moskova”
suriye	şam	“suriye” AND “ şam”

Table 3.2: Sample queries for answer pattern extraction

3.4 Selecting Sentences

In order to extract answer patterns, the content of each document is broken into sentences. Answer patterns are regular expressions representing the words and punctuation that occur between and around the question and answer phrases. So, only the sentences which contain both phrases are used to extract answer patterns. Other sentences that do not contain both phrases are ignored.

3.5 Identifying Answer Pattern Boundaries

After the sentences containing the question and answer phrases are selected, the boundaries of the regular expressions are identified. In this step, the words and punctuation between and around the question and answer phrases are identified as answer pattern boundaries. An answer pattern can be in one of the following four forms:

- <Q><intermediate string><A>
- <A><intermediate string><Q>
- <Q><intermediate string><A><boundary string>

- <boundary string><A><intermediate string><Q>

Here, <Q> stands for the question phrase and <A> stands for the potential answer. *Boundary string* is used in the last two forms to identify the boundary of answer.

The followings are two example sentences. For these examples, question phrase is *Türkiye*, answer phrase is *Ankara* and question type is CAPITAL-OF-COUNTRY.

(1) "Asya ve Avrupa kıtalarını birbirine bağlayan yollar üzerinde bulunan *Türkiye*'nin başkenti olan *Ankara* şehri Anadolu'nun merkezinde yer alır."

(2) "Başkent *Ankara*, *Türkiye*'nin ikinci büyük şehridir."

Following answer pattern boundaries are identified.

- An answer pattern covers the question phrase, answer phrase and an arbitrary string in between these phrases.

  (1.1) "Türkiye'nin başkenti olan Ankara"

  (2.1) "Ankara, Türkiye"

- An answer pattern covers the question phrase, answer phrase, an arbitrary string in between these phrases plus one token following the answer phrase to indicate where it ends.

  (1.2) "Türkiye'nin başkenti olan Ankara şehri‘"

- An answer pattern covers the question phrase, answer phrase, an arbitrary string in between these phrases plus one token preceding the answer phrase to indicate where it starts.

  (2.2) "Başkent Ankara, Türkiye"

3.6 Replacing Question and Answer Phrases

In this step in order to generalize the answer patterns, question phrase and answer phrase are replaced with the tags <Q> and <A> respectively. In the following examples, the question phrase "türkiye" is replaced by <Q> tag and the answer phrase "ankara" is replaced by <A> tag.

- "<Q>'nin başkenti olan <A>"
- "<A>, <Q>"
- "<Q>'nin başkenti olan <A> şehri"
- "başkent <A>, <Q>"

3.7 Building Regular Expressions

Answer patterns are extracted by applying different methods. Raw String methods do not change the strings. Stemmed String methods stem the words in the strings before building regular expressions. Named Entity Tagged String methods replace the words in the string with their named entity tags. Stemmed String and Named Entity Tagged String methods extract more general answer patterns while Raw String methods extract more specific answer patterns. After a method is applied, the corresponding regular expression is built for that answer pattern by replacing <A> tag with "(.*?)". When an answer pattern regular expression matches a sentence, the string in place of "(.*?)" is extracted as an answer. The details of answer pattern extraction methods are given in Chapter 4.

Each answer pattern has a confidence factor. The reliability of an answer pattern is determined by means of its confidence factor value. Confidence factors of all newly extracted answer patterns are set to zero initially. Confidence factors are updated in the second phase of the answer pattern learning process. If an answer pattern never matches and never extracts an answer in the second phase

of the learning process, the confidence factor remains zero. The answer patterns whose confidence factor is zero are eliminated at the end of the learning process.

If an answer pattern matches and extracts an answer in the second phase of the learning process, its confidence factor is updated according to the correctness of the produced answer. While the extracted answers are correct, the confidence factor of the answer pattern increases. While the extracted answers are incorrect, the confidence factor of the answer pattern decreases. The details of the confidence factor assignment are presented in Chapter 5.

Chapter 4

Answer Pattern Extraction Methods

Answer patterns can be extracted using five different methods. Answer pattern extraction methods are applied after the boundary is determined. The methods are explained in the following sections.

4.1 Method 1: Raw String

After the boundary of an answer pattern is determined, only the question and answer phrases are replaced by <Q> and <A> tags respectively and all the other parts of the answer pattern remain the same. In Table 4.1, some sample answer pattern strings are given in the left column after their boundaries are identified. Question phrases and answer phrases are shown as underlined. Answer patterns extracted by Raw String method are given in the right column.

This method extracts surface level answer patterns. Since the answer pattern extracted by Raw String method contains the surface form of words, the extracted answer patterns by Raw String method are specific. Since this method does not use any special NLP technique such as stemming and named entity tagging, the

Answer Pattern String	Answer Pattern
Türkiye'nin başkenti Ankara	<Q>'nin başkenti <A>
İnce Memed romanının yazarı Yaşar Kemal	<Q> romanının yazarı <A>
Mustafa Kemal Atatürk 1881 yılında	<Q> <A> yılında
dili Türkçe olan Türkiye	dili <A> olan <Q>

Table 4.1: Some sample answer patterns extracted by Raw String method

usage of these patterns will be fast during question answering.

4.2 Method 2: Raw String with Answer Type

After Raw String method is applied, the answer type (named entity tag of the answer) is added to the answer patterns extracted by Raw String method. Answer type is identified according to question type. As explained in Chapter 3, question type is given as input to the system along with question-answer pairs of that question type. In Table 4.2, answer patterns that are extracted by Raw String method are shown in the left column and answer patterns that are extracted by this method are shown in the right column.

Answer Pattern (Raw String)	Answer Pattern (with Answer Type)
<Q>'nin başkenti <A>	<Q>'nin başkenti <A-NECity>
<Q> romanının yazarı <A>	<Q> romanının yazarı <A-NEPersonName>
<Q> <A> yılında	<Q> <A-NEDate> yılında
dili <A> olan <Q>	dili <A> olan <Q>

Table 4.2: Some sample answer patterns extracted by Raw String with Answer Type method

If the answer type for a question type is not identified, new answer patterns cannot be extracted by this method. Since the answer type of the fourth question is not identified, the answer pattern is the same as the answer pattern produced by Raw String method.

During question answering, if the answer pattern matches a sentence and a

candidate answer is extracted, the named entity tag of the candidate answer is determined by using a Named Entity Tagger. If its named entity tag is the same as the expected answer type, then the answer is produced. If its named entity tag does not match, no answer is produced.

Since the answer pattern extracted by this method contains the surface form of words and the expected answer type, the extracted answer patterns are more specific. This yields that the confidence factors of the answer patterns learned by this method are higher than the answer patterns learned by Raw String method.

We use a Turkish Named Entity Tagger which was developed previously. This method requires to tag all the words in the sentences so the processing time for question answering will be longer than the Raw String method.

4.3 Method 3: Stemmed String

After the boundary of an answer pattern is determined, all of the words in the boundary are stemmed. The goal of this method is to remove all affixes of the words and then leave only the stems of the words. In Table 4.3, same sample sentences are given in the left column after their boundaries are identified. Question phrases and answer phrases are shown as underlined. Answer patterns extracted by Stemmed String method are given in the right column.

Answer Pattern String	Answer Pattern
Türkiye'nin başkenti Ankara	<Q> başk <A>
İnce Memed romanının yazarı Yaşar Kemal	<Q> roma yaza <A>
Mustafa Kemal Atatürk 1881 yılında	<Q> <A> yılı
dili Türkçe olan Türkiye	dili <A> olan <Q>

Table 4.3: Some sample answer patterns extracted by Stemmed String method

We use the cut off technique for stemming. The first four characters in the words are remained and the other characters are removed. This method requires to stem all the words in the sentences so the processing time for question answering is longer than the Raw String method.

4.4 Method 4: Stemmed String with Answer Type

After Stemmed String method is applied, the answer type (named entity tag of the answer) is added to the answer patterns extracted by Stemmed String method. Answer type is identified according to question type. As explained in Chapter 3, question type is given as input to the system along with question-answer pairs of that question type. In Table 4.4, answer patterns that are extracted by Stemmed String method are given in the left column and answer patterns that are extracted by this method are shown in the right column.

Answer Pattern (Stemmed String)	Answer Pattern(with Answer Type)
<Q> başk <A>	<Q> başk <A-NECity>
<Q> roma yaza <A>	<Q> roma yaza <A-NEPersonName>
<Q> <A> yılı	<Q> <A-NEDate> yılı
dili <A> olan <Q>	dili <A> olan <Q>

Table 4.4: Some sample answer patterns extracted by Stemmed String with Answer Type method

If the answer type for a question type is not identified, new answer patterns cannot be extracted by this method. Since the answer type of the fourth question is not identified, the answer pattern is the same as the answer pattern produced by Stemmed String method.

During question answering, if the answer pattern matches a sentence and a candidate answer is extracted, the named entity tag of the candidate answer is determined by using Turkish Named Entity Tagger. If its named entity tag is the same as the expected answer type, then the answer is produced. If its named entity tag does not match, no answer is produced.

Since the answer pattern extracted by this method contains the expected answer type, the extracted answer patterns are more specific. This yields that the confidence factors of the answer patterns learned by this method are higher than the answer patterns learned by Stemmed String method.

4.5 Method 5: Named Entity Tagged String

After the boundary of an answer pattern is determined, the named entity tags of all the words are assigned by Turkish Named Entity Tagger. Then, the words are replaced by the names of their named entity tags. For instance, if the named entity tag of a word is City, the word is replaced by the string "NE_City", if the named entity tag of a word is Date, the word is replaced by the string "NE_Date", etc. If the consecutive words have the same named entity tag, they are replaced by only one named entity tag. The words whose named entity tags cannot be identified are not replaced and used as they are.

For example, string (1) is the answer pattern string after its boundary is determined and string (2) is the answer pattern extracted by Named Entity Tagged String method.

(1) "Mustafa Kemal Atatürk (d. 19 Mayıs 1881, Selanik - ö. 10 Kasım 1938, İstanbul"

(2) "<Q> (d. NE_Date, NE_City - ö. NE_Date, <A_NECity>"

All sentences are tagged before question answering. If an answer pattern matches a tagged sentence, an answer is extracted from that sentence.

This method requires to tag all the words in the sentences so the processing time for question answering is longer than the Raw String method.

Chapter 5

Confidence Factor Assignment

In this chapter, the second phase of answer pattern learning process is explained. The goal of the second phase is to assign a confidence factor to each answer pattern that is extracted in the first phase. At the end of the phase, the answer patterns whose confidence factors are under a given threshold are eliminated.

Each answer pattern has a confidence factor. The reliability of an answer pattern is determined by means of its confidence factor value. Confidence factor of an answer pattern is similar to precision of that answer pattern. To assign confidence factor, two attributes are used:

N_{TRUE} **:** Number of times that the answer pattern matches a sentence and the extracted answer is correct.

N_{FALSE} **:** Number of times that the answer pattern matches a sentence and the extracted answer is incorrect.

Each answer pattern has its own N_{TRUE} and N_{FALSE} attributes. Following formula is used to update the confidence factors. We use add-one smoothing technique in the formula to penalize the answer patterns which have small number of correct matches.

$$ConfidenceFactor = (N_{TRUE} + 1)/(N_{TRUE} + N_{FALSE} + 2) \quad (5.1)$$

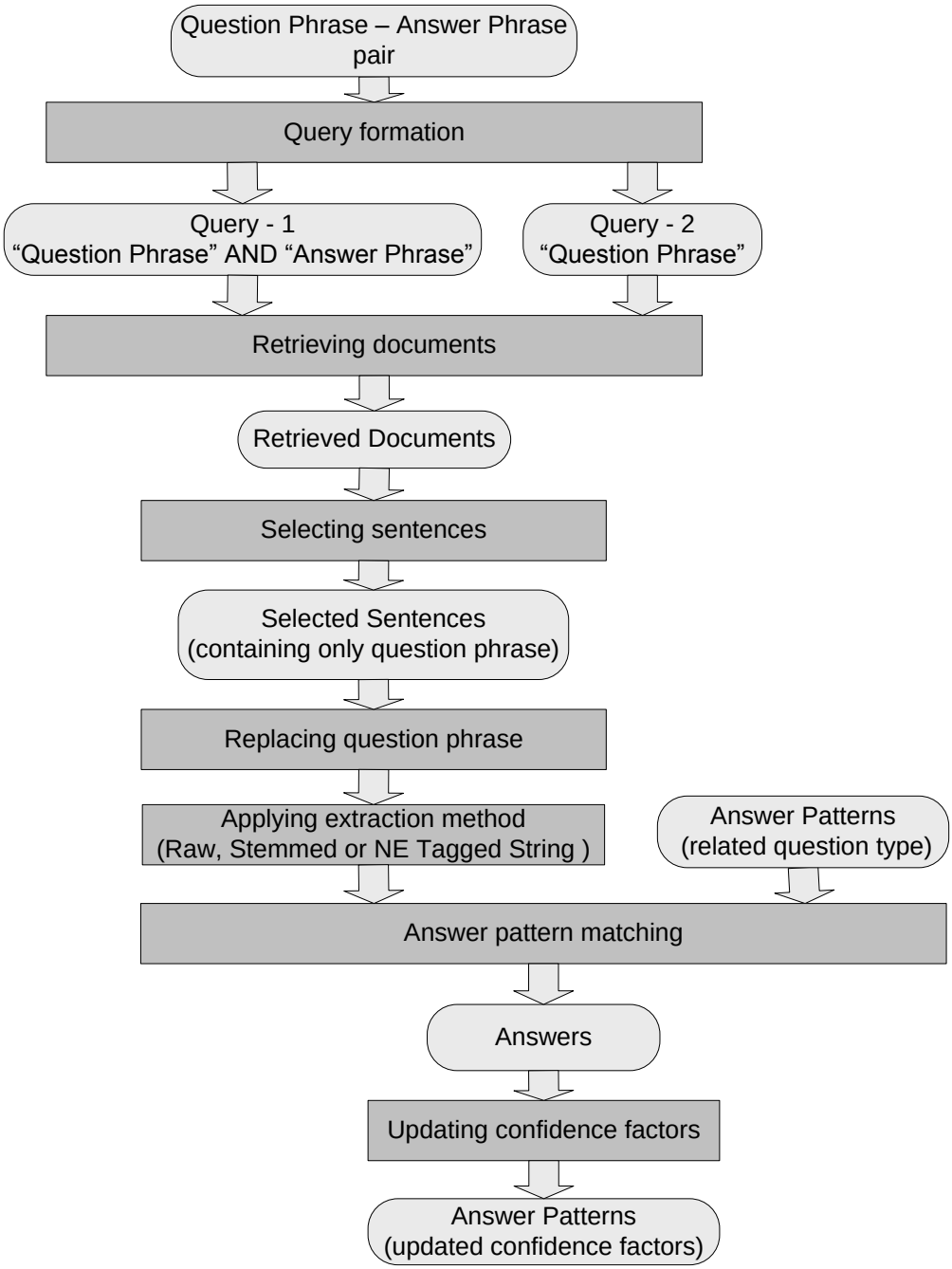


Figure 5.1: Confidence factor assignment process

Figure 5.1 shows the steps of the confidence factor assignment process. Each

step is represented by a rectangle and the input and/or output of a step is represented by a rounded box. The steps of the process are explained in detail in the following sections.

5.1 Preparing a Set of Question-Answer Pairs

A set of question-answer pairs is prepared for each question type. The set is prepared manually and all pairs have to be correct. As an example, the set used for CAPITAL-OF-COUNTRY question type is given in Table 5.1 which contains the same pairs given in Chapter 3. Each line in the table contains a question-answer pair.

Question Phrase	Answer Phrase
türkiye	ankara
fransa	paris
almanya	berlin
bulgaristan	sofya
yunanistan	atina
romanya	bükreş
ingiltere	londra
çin	pekin
rusya	moskova
suriye	şam

Table 5.1: Sample question-answer pairs for confidence factor assignment

The same set of question-answer pairs is used by both phases of the learning process.

5.2 Querying the Web

Two queries are formulated in this step. The first query is formed by using only question phrase. The second query is formed by using both question phrase and

answer phrase (question phrase and answer phrase are AND'ed). Table 5.2 shows the queries formed for the sample pairs.

Question	Answer	First Query	Second Query
türkiye	ankara	"türkiye"	"türkiye" AND "ankara"
fransa	paris	"fransa"	"fransa" AND "paris"
almanya	berlin	"almanya"	"almanya" AND "berlin"
bulgaristan	sofya	"bulgaristan"	"bulgaristan" AND "sofya"
yunanistan	atina	"yunanistan"	"yunanistan" AND "atina"
romanya	bükreş	"romanya"	"romanya" AND "bükreş"
ingiltere	londra	"ingiltere"	"ingiltere" AND "londra"
çin	pekin	"çin"	"çin" AND "pekin"
rusya	moskova	"rusya"	"rusya" AND "moskova"
suriye	şam	"suriye"	"suriye" AND " şam"

Table 5.2: Sample queries for confidence factor assignment

Web Search Engine returns a ranked list of the related web pages. Since we also use the contents of the retrieved web pages for this phase, the retrieved web pages are downloaded.

The retrieved web pages by the first query contain only the question phrase whereas the retrieved web pages by the second query contain both question phrase and answer phrase. If only the first query is searched, the returned documents do not usually contain the answer phrase. So the second query is also used. Using only the second query can cause to favor some answer patterns. Whenever these answer patterns match, the produced answer is generally correct because the query also contains the answer phrase. To assure the balance, two queries are formulated. Half of the document set is composed of the documents which are retrieved by the first query and the other half is composed of the documents which are retrieved by the second query.

5.3 Selecting Sentences

The content of each document is broken into sentences. Sentence selection is different from the first phase. The sentences which contain only the question

phrase are used for this phase. The aim of selecting sentences which contain only the question phrase is to prevent selecting only the sentences containing the correct answer phrase. For instance, the following sentences are selected for a question-answer pair whose question phrase is "türkiye" and answer phrase is "ankara".

(1) "13 Ekim 1923 tarihinde Türkiye'nin başkenti Ankara olarak ilan edilmiştir."

(2) "Türkiye'nin başkenti ve ikinci büyük şehri olan Ankara'nın ilk yerleşim tarihi tam olarak bilinmemektedir."

(3) "Türkiye başbakanının yarın Davos'ta olması bekleniyor."

(4) "Türkiye'nin başkenti susuzluk tehlikesi ile karşı karşıya."

Sentences (1) and (2) contain both the question phrase and answer phrase, but sentences (3) and (4) contain only the question phrase. If we select the sentences which contain both question phrase and answer phrase, sentences (3) and (4) are not selected.

5.4 Replacing Question Phrase

Question phrases in the selected sentences are replaced by <Q> tag to generalize the sentences. In the following sentences, the question phrase "türkiye" is replaced by <Q> tag.

(1) "13 Ekim 1923 tarihinde <Q>'nin başkenti Ankara olarak ilan edilmiştir."

(2) "<Q>'nin başkenti ve ikinci büyük şehri olan Ankara'nın ilk yerleşim tarihi tam olarak bilinmemektedir."

(3) "<Q> başbakanının yarın Davos'ta olması bekleniyor."

(4) "<Q>'nin başkenti susuzluk tehlikesi ile karşı karşıya."

5.5 Updating Confidence Factors

If an answer pattern matches a sentence, its confidence factor is updated according to the correctness of the answer. If the extracted answer is correct (same as the given answer), confidence factor of the answer pattern increases. If the extracted answer is incorrect (different from the given answer), confidence factor of the answer pattern decreases.

Assume that the following answer pattern which is created by Raw String method is used to match the example sentences given above.

"<Q>'nin başkenti <A>"

Table 5.3 shows the match status for this answer pattern. The answer pattern matches sentences (1), (2) and (4). The extracted answer from sentence (1) is correct so the confidence factor of the answer pattern increases. The extracted answers from sentences (2) and (4) are incorrect so the confidence factor of the answer pattern decreases. Thus, the confidence factor of the answer pattern will be 2/5 according to Formula 5.1. ($N_{TRUE} = 1$ and $N_{FALSE} = 2$)

Sentence	Match	Answer	Correctness	Confidence Factor
(1)	Match	"ankara"	Correct	Increases
(2)	Match	"ve"	Incorrect	Decreases
(3)	No Match	-	No Change	No Change
(4)	Match	"susuzluk"	Incorrect	Decreases

Table 5.3: Extracted answers by an answer pattern created by Raw String method

Assume that the following answer pattern which is created by Raw String with Answer Type method is used to match the example sentences given above.

"<Q>'nin başkenti <A-NECity>"

Table 5.4 shows the match status for this answer pattern. The answer pattern matches only sentence (1) and the extracted answer from sentence (1) is correct so the confidence factor of the answer pattern increases. The confidence factor of the answer pattern will be 2/3 according to Formula 5.1. ($N_{TRUE} = 1$ and

$N_{FALSE} = 0$)

Sentence	Match	Answer	Correctness	Confidence Factor
(1)	Match	"ankara"	Correct	Increases
(2)	Match	-	No Change	No Change
(3)	No Match	-	No Change	No Change
(4)	Match	-	No Change	No Change

Table 5.4: Extracted answers by an answer pattern created by Raw String with Answer Type method

5.6 Eliminating Unreliable Answer Patterns

After all pairs are processed, the answer patterns whose confidence factors are under a certain threshold are eliminated. These patterns are considered as unreliable and the possibility of producing correct answer is very low so these answer patterns are not applied to the passages in question answering.

The goal of this phase is to eliminate the answer patterns which are unlikely to produce correct answers. After this phase is completed, a set of answer patterns are learned for each question type and they are ready to be used in Answer Processing phase of question answering.

Chapter 6

Question Answering using Answer Pattern Matching

After answer patterns are learned, they are used to extract answers. The base algorithm used to extract answers is as follows:

1. The system takes question phrase and question type as input.
2. A query is created from the question phrase and the query is submitted to a search engine.
3. The top N retrieved documents are examined. They are segmented into sentences and the sentences containing question phrase are selected.
4. Answer patterns of the given question type are applied to the selected sentences to extract answers.
5. If an answer pattern matches a sentence, an answer is extracted from that sentence. The extracted answer can be expanded according to its named entity tag.
6. The confidence value of the answer is assessed by the confidence factor of the matched answer pattern.

7. The answer is added to the candidate answers list. The candidate answers list is sorted according to the confidence values.

Our base algorithm uses only the question phrase to create a query. Since the created query is a general query, the retrieved documents may be insufficient to find the answer. So, we extend our base algorithm to retrieve documents that are more likely to contain answer. Our approach is based on query expansion. We use the most reliable answer patterns to extend the queries. The base algorithm is detailed in Section 6.1 and our query expansion approach is defined in Section 6.2.

In the last section, re-ranking of the returned answers method is explained. Answers in the candidate answers list are re-sorted according to their frequency count. Frequency count method is defined in Chapter 1. The application of frequency counting method in our system is explained in Section 6.3.

6.1 Question Answering without Query Expansion

In this section, our base algorithm for factoid QA is detailed. Our system architecture is similar to the typical factoid QA system architecture defined in Chapter 1. Figure 6.1 shows the phases and the tasks performed in each phase. The first phase is Question Processing phase and two tasks are performed by this phase. One of them is question type identification and the other task is query formation. Typical QA systems take questions as natural language question sentences. They identify the question type and use the words in the question sentence to formulate query(ies). Some of the methods used for question type identification and query formation are explained in Chapter 1. In our approach, each question is expressed as a question phrase instead of a question sentence. The question type is also given as an input to the system so question type identification becomes unnecessary. The system takes *question phrase-answer phrase* pairs while learning answer patterns. Similar to learning answer patterns, the system takes *question phrases* during question answering. In query formation, only the question phrase

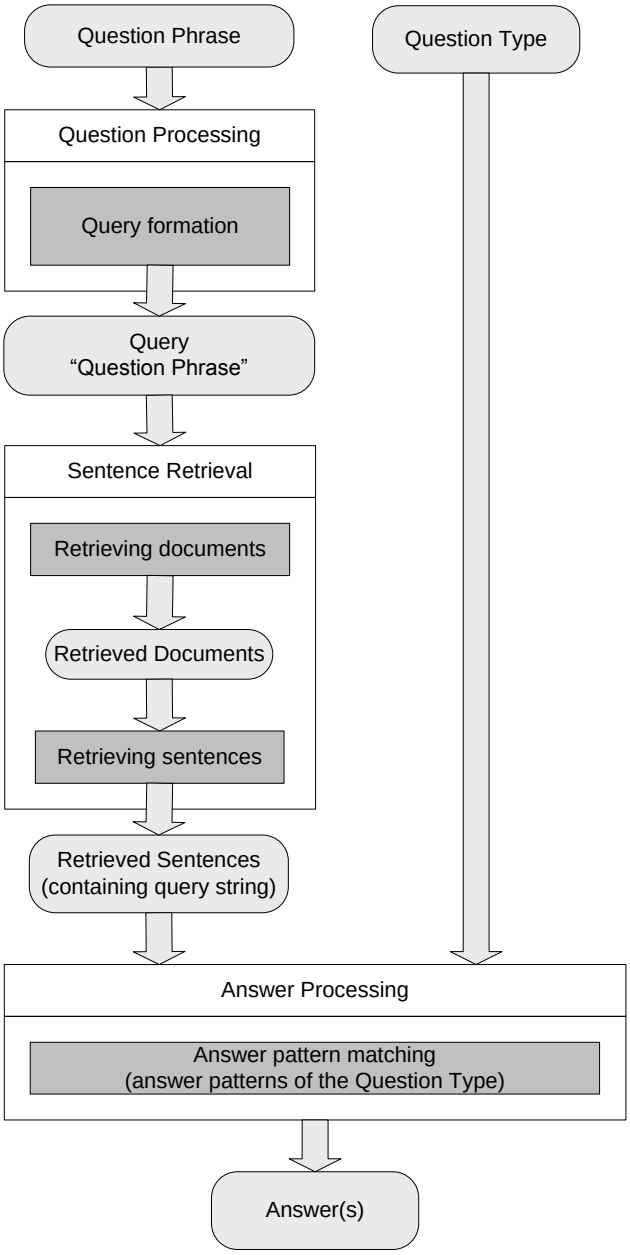


Figure 6.1: Factoid question answering without query expansion

is used to form a query. For example, the query for a CAPITAL-OF-COUNTRY question type is formulated using the name of the country, the query for a DATE-OF-BIRTH question type is formulated using the name of the person, etc. Some example question phrases and their queries are given in Table 6.1.

Question Type	Question Phrase	Query
Capital-of-Country	Danimarka	“danimarka”
Date-of-Birth	Mustafa Kemal Atatürk	“mustafa kemal atatürk”
Place-of-Birth	Mustafa Kemal Atatürk	“mustafa kemal atatürk”
Author	İnce Memed	“ince memed”
Actor	Eşkiya	“eşkiya”

Table 6.1: Some question phrases and their queries

Queries are passed as input to Sentence Retrieval phase. Queries are submitted to a search engine. We use Bing Web Search Engine [3]. The top 250 web documents retrieved by the search engine are downloaded. Each document is segmented into sentences. The sentences which do not contain question phrase are ignored. The sentences containing question phrase are kept. Then, question phrases in the sentences are replaced by <Q> tag. The replaced sentences are returned to Answer Processing phase.

Answer patterns of the question type are applied to the returned sentences to extract answers in Answer Processing phase. Preprocessing of sentences may be required according to the answer pattern extraction method (stemming or named entity tagging).

- If the applied answer pattern is learned by Raw String method, the returned sentences are used directly. (No preprocessing is required.)
- If the applied pattern is learned by Stemmed String method, the words in the sentences are first stemmed. The words in the sentences are replaced with their stems. Then, the stemmed versions of the sentences are used.
- If the applied answer pattern is learned by Named Entity Tagged String method, the sentences are first named entity tagged. The words in the sentences are replaced with their associated NE tags. If consecutive words have the same named entity tag, all of them are replaced with only one NE tag. Then, the named entity tagged versions of the sentences are used.

For example, if answer patterns learned by Stemmed String method are applied, sentence (2) is used instead of sentence (1). (Sentence (2) is the stemmed version of sentence (1).) If answer patterns learned by NE Tagged String method are applied, sentence (4) is used instead of sentence (3). (Sentence (4) is the NE tagged version of sentence (3).)

(1) "<Q>'nin doğum tarihi 24 Mayıs 1953."

(2) "<Q> doğu tari 24 Mayı 1953."

(3) "<Q> (d. 19 Mayıs 1881, Selanik - ö. 10 Kasım 1938, İstanbul)."

(4) "<Q> (d. NE_Date, NE_City - ö. NE_Date, _NECity)."

If an answer pattern matches a sentence, an answer is extracted from that sentence. The extracted answer can be expanded according to its named entity tag. If the extracted answer has a named entity tag and the words around the extracted answer have the same named entity tag, the answer is expanded by adding these words. Then, the answer is added to the candidate answers list. The confidence value of the answer is assessed by the confidence factor of the matched answer pattern. The candidate answers list is sorted according to the confidence values.

6.2 Question Answering with Query Expansion

Our base algorithm creates a query for a question phrase and the query contains only the question phrase. The retrieved documents by the query may be insufficient to extract the answer because the query is too general. We develop an approach to extend our base algorithm. The goal of the approach is to retrieve documents that are more likely to contain answer. Our approach is based on query expansion. We use the most reliable answer patterns to extend the queries. In this section, we explain how answer patterns are used for query expansion.

Query Expansion is the process of reformulating a query. The goal of Query

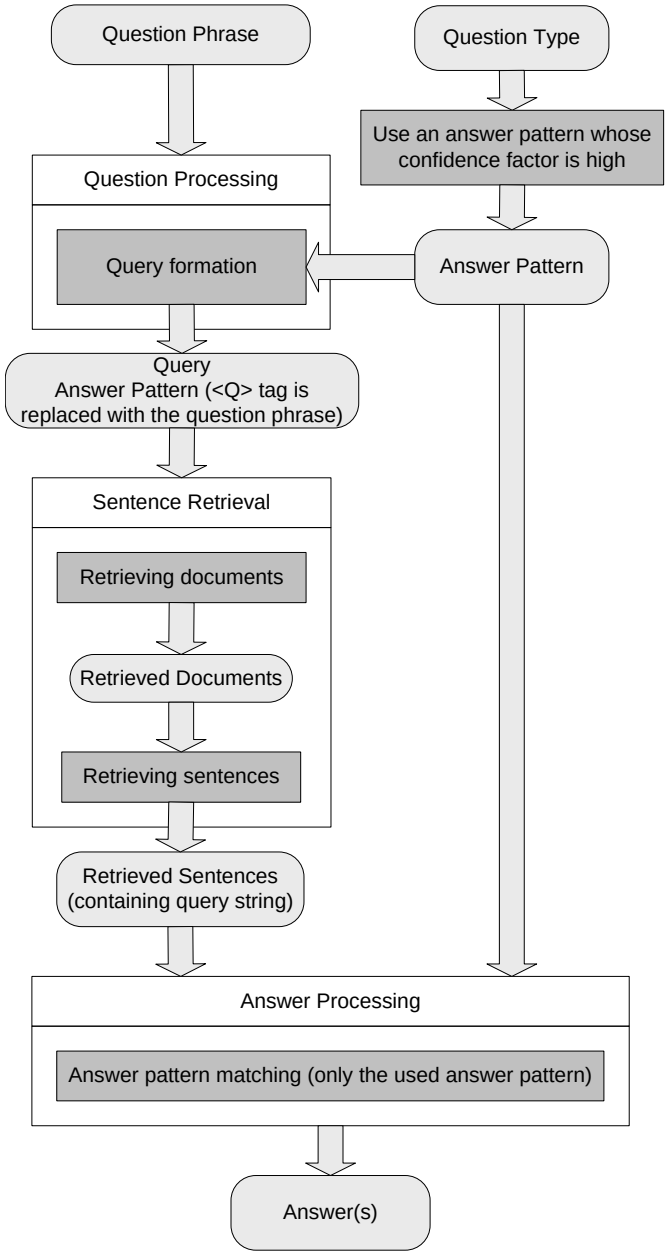


Figure 6.2: Factoid question answering with query expansion

Expansion is to improve retrieval performance. Query Expansion involves different techniques such as adding synonyms of words to the query, adding different morphological forms of words to the query, etc.

We use answer patterns to extend queries. After answer patterns are learned, reliable answer patterns are determined. The more the confidence factor of an

answer pattern is, the more the answer pattern is reliable. We can use these reliable answer patterns to extend our queries. Answer patterns are regular expressions containing <Q> tag in place of question phrases. Question phrases are replaced with this tag in learning phase. This time, <Q> tag is replaced with the question phrase to create queries.

Some example answer patterns for CAPITAL-OF-COUNTRY question type are given in the first column of Table 6.2. For example, if question phrase is "Danimarka", <Q> tags in the answer patterns are replaced with the question phrase "danimarka". Created queries are shown in the second column.

Answer Pattern	Query
<Q>'nin başkenti <A>	"danimarka'nin başkenti"
<Q>'nın başkenti <A>	"danimarka'nın başkenti"
<Q>'nin başkenti olan <A>	"danimarka'nin başkenti olan"
<Q> başkenti <A>	"danimarka başkenti"

Table 6.2: Some sample queries created by using answer patterns

Figure 6.2 shows how an answer pattern is used for query expansion. The confidence factor of the answer pattern should be high enough to be used for query expansion. Question phrase and the answer pattern are the inputs of Query Formation task. A query is built by using these two inputs. <Q> tag is replaced with the question phrase and a query is created. The query is submitted to the search engine and selected sentences are returned by Sentence Retrieval phase. If a sentence contains the question phrase, it is selected and it is added to the returned sentence list. Selected sentences and the answer pattern used for query expansion are given to Answer Processing phase as input. Only the answer pattern used for query expansion is applied to the selected sentences. Other answer patterns of the question type are not applied. If the answer pattern matches a sentence, an answer is extracted from that sentence as explained in Section 6.1. Then, the answer is added to the candidate answers list. The confidence value of the answer is assessed by the confidence factor of the matched answer pattern.

Answer patterns whose confidence factor are high enough are used for query expansion one by one. A query without query expansion is also used to find

answers as explained in Section 6.1. The answers produced without query expansion and the answers produced with query expansion are sorted according to their confidence values.

6.3 Answer Re-ranking Using Frequency Counting

We use two answer ranking approaches:

1. The first ranking approach is based on only confidence values assigned to answers.
2. The second ranking approach is based on both confidence values and Frequency Counting. (Refer to Chapter 1)

In our first approach, after candidate answers are extracted by using answer pattern matching, candidate answers are sorted according to their *confidence values*. Confidence value of an answer is assigned as the confidence factor of the answer pattern which extracts the answer.

In our second approach, after candidate answers are extracted by using answer pattern matching, candidate answers are sorted according to their *total confidence values*. When a new candidate answer is extracted, it is searched in the candidate answers list. If the same candidate answer is already included in the list, its confidence value is increased by adding the confidence factor of the new candidate answer. The more the same answer is extracted, the more its total confidence value increases. The second approach relies on correct answers to appear more frequently than other incorrect answers.

Chapter 7

System Evaluation and Results

Several experiments have been performed to evaluate the performance of the system. In the first section, the metrics used for system evaluation are explained. Results and findings from different answer pattern extraction methods are provided in the second section. The evaluation of answer re-ranking approach is given in the third section. The effect of query expansion on the system is discussed in the fourth section. The last section compares the performance of the system with the other question answering systems.

Seven question types are used for evaluation. These question types are *Author*, *Capital*, *DateOfBirth*, *DateOfDeath*, *Language*, *PlaceOfBirth*, and *PlaceOfDeath*. Answer patterns are learned for each question type by using 15 question-answer pairs. The system is evaluated by using another 15 question-answer pairs. Question-Answer pairs are given in Appendix A. Learned answer patterns are given in Appendix C.

7.1 Evaluation Metrics

The following metrics are based on the first answer returned by question answering system. The first index is the most important index because it will be the

answer that is produced by the system. The correct answers should have as low index as possible.

Number of test questions represents the number of questions in the test set.

Number of returned answers represents the number of questions that the system has returned some answers for them.

Number of correct answers represents the number of questions that the system has returned correct answer for them.

Precision is *number of correct answers / number of returned answers.*

Recall is *number of correct answers / number of test questions.*

MRR Mean Reciprocal Rank (MRR) considers the rank of the first correct answer in a list of possible answers. MRR score for an individual question is the reciprocal of the rank at which the first correct answer is returned or 0 (zero) if no correct answer is returned. For instance, if a QA system returns the correct answer in the first place, the MRR value of the system is %100 or 1. If the correct answer is in the fourth place, the MRR value is %25 or 0.25. *MRR* metric for a QA system represents the mean over the set of questions in the test and it is bounded between 0 and 1, inclusive. *MRR* metric was used to evaluate QA systems at TREC [24] [25].

F-measure is *2 * (precision * recall) / (precision + recall).*

Number of answer patterns represents the number of answer patterns used to extract answers.

7.2 Evaluation of Answer Pattern Extraction Methods

We evaluate answer pattern extraction methods in this section. First, the results and findings of each answer pattern extraction method are given and the results

of the combination of different answer pattern extraction methods are provided. The confidence factor threshold for answer patterns is selected as 0.75 in the evaluations. Then, the effect of confidence factor threshold on the performance of the system is discussed. Results of answer pattern extraction methods at different thresholds are given in Appendix B.

7.2.1 Method 1: Raw String

We give the results of Raw String method in this section. The system uses only the answer patterns learned by Raw String method for this evaluation. The results are shown in Table 7.1. In Table 7.1, $\#Q$ means the number of test questions, $\#A$ means the number of returned answers, $\#CA$ means the number of correct answers in the first position, and $\#AP$ means the number of answer patterns. The first column shows the question types and the last row shows the total results. We use the same table template in the presentation of the results of the other answer pattern extraction methods.

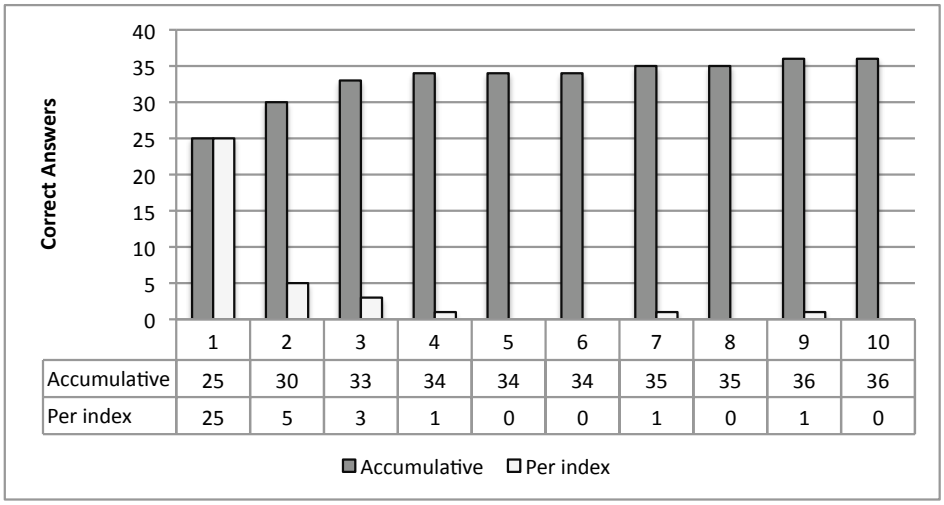
Question Type	#Q	#A	#CA	MRR	Precision	Recall	#AP
Author	15	9	1	0.19	0.11	0.07	241
Capital	15	14	12	0.84	0.86	0.80	1181
DateOfBirth	15	0	0	0.00	0.00	0.00	30
DateOfDeath	15	1	1	0.07	1.00	0.07	183
Language	15	14	9	0.70	0.64	0.60	597
PlaceOfBirth	15	5	1	0.07	0.20	0.07	370
PlaceOfDeath	15	1	1	0.07	1.00	0.07	467
TOTAL	105	44	25	0.28	0.57	0.24	3069

Table 7.1: Results of Raw String method

According to Table 7.1, the best results are obtained for *Capital* question type and the worst results are obtained for *DateOfBirth* question type. Number of answer patterns learned for *Capital* question type is much more than the number of answer patterns learned for *DateOfBirth* question type.

Figure 7.1 shows the number of correct answers returned between the indices

1 - 10. According to Figure 7.1, most of the correct answers are returned at the first index, however, some correct answers are returned at the second and the third indices. The *Recall* value increases to 0.31 (33/105) from 0.24 if the number of correct answers are based on the first three answers returned rather than the first answer returned.



	1	2	3	4	5	6	7	8	9	10
Accumulative	25	30	33	34	34	34	35	35	36	36
Per index	25	5	3	1	0	0	1	0	1	0

Figure 7.1: Correct answers returned by Raw String method

7.2.2 Method 2: Raw String with Answer Type

We present the results of Raw String with Answer Type method in this section. The system uses only the answer patterns learned by Raw String with Answer Type method for this evaluation. The results are shown in Table 7.2.

The results of *Capital* question type are the best results and the results of *DateOfBirth* question type are the worst results according to Table 7.2. Number of answer patterns of a question type affects the performance of the system for that question type. There is no change in the results of *Language* question type because *Language* question type does not have an answer type.

Figure 7.2 shows the number of correct answers returned between the indices 1 - 10. According to Figure 7.2, most of the correct answers are returned at the first index, a few correct answers are returned at the second and the third

Question Type	#Q	#A	#CA	MRR	Precision	Recall	#AP
Author	15	6	6	0.40	1.00	0.40	241
Capital	15	13	13	0.87	1.00	0.87	919
DateOfBirth	15	0	0	0.00	0.00	0.00	30
DateOfDeath	15	1	1	0.07	1.00	0.07	183
Language	15	14	9	0.70	0.64	0.60	597
PlaceOfBirth	15	1	1	0.07	1.00	0.07	273
PlaceOfDeath	15	1	1	0.07	1.00	0.07	327
TOTAL	105	36	31	0.31	0.86	0.30	2570

Table 7.2: Results of Raw String with Answer Type method

indices. There is only a small increase in the *Recall* value if the number of correct answers are based on the first three answers returned rather than the first answer returned.

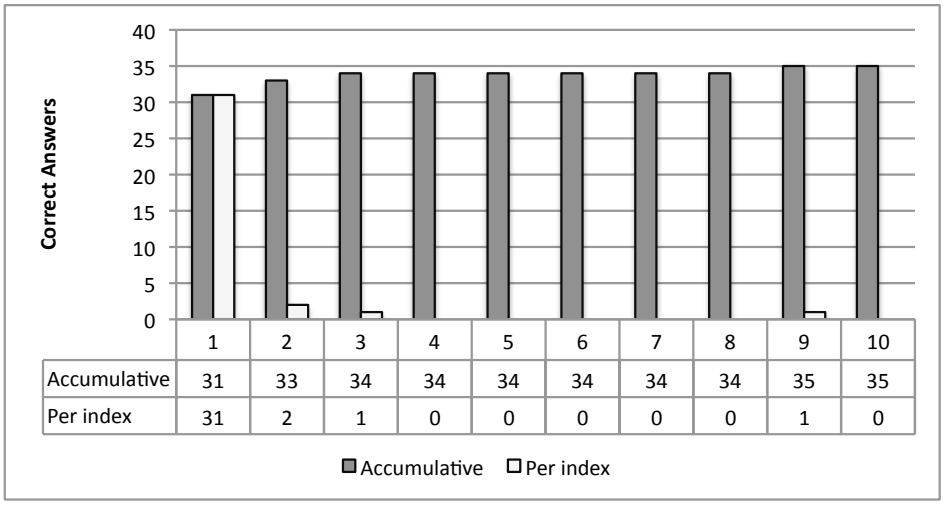


Figure 7.2: Correct answers returned by Raw String with Answer Type method

Figure 7.3 shows the effect of answer type checking for Raw String methods. Here *Raw* means Raw String method, and *RawNE* means Raw String with Answer Type method. *MRR*, *Recall* and *Precision* values of Raw String with Answer Type method are greater than *MRR*, *Recall* and *Precision* values of Raw String method as shown in Figure 7.3. Especially, a significant increase occurs in *Precision* value. So, checking the answer type of a candidate answer before returning it as an answer increases the performance of the system. For instance,

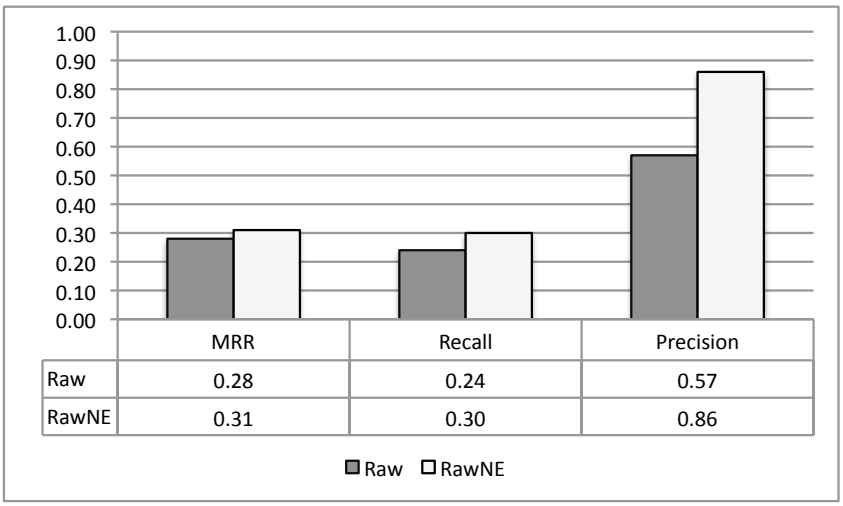


Figure 7.3: Effect of answer type checking for Raw String methods

one of the test questions is "Danimarka" for *Capital* question type. The system returns a wrong answer at the first index for that question if the system uses only the answer patterns learned by Raw String method. The returned answer is "olan" and it is extracted by "<Q>'nın başkenti <A>" answer pattern from the following sentence.

"Sonunda Kral I. Valdemar (1131 - 1182) Danimarka'yı tekrar birleştirmeyi başardı ve günümüzde Danimarka'nın başkenti olan Kopenhag'ın temellerini attı."

The system returns a correct answer at the first index for that question if the system uses the answer patterns learned by Raw String with Answer Type method. The returned answer is "Kopenhag'ın" and it is extracted by "<Q>'nın başkenti olan <A-NECity>" answer pattern from the same sentence.

7.2.3 Method 3: Stemmed String

We give the results of Stemmed String method in this section. The system uses only the answer patterns learned by Stemmed String method for this evaluation. The results are shown in Table 7.3. The best results are obtained for *Capital* question type and the worst results are obtained for *DateOfBirth* question type

according to Table 7.3.

Question Type	#Q	#A	#CA	MRR	Precision	Recall	#AP
Author	15	10	1	0.16	0.10	0.07	140
Capital	15	15	14	0.93	0.93	0.93	947
DateOfBirth	15	0	0	0.00	0.00	0.00	26
DateOfDeath	15	3	1	0.07	0.33	0.07	154
Language	15	13	9	0.69	0.69	0.60	551
PlaceOfBirth	15	4	1	0.07	0.25	0.07	272
PlaceOfDeath	15	2	1	0.10	0.50	0.07	388
TOTAL	105	47	27	0.29	0.57	0.26	2478

Table 7.3: Results of Stemmed String method

Figure 7.4 shows the number of correct answers returned between the indices 1 - 10. According to Figure 7.4, most of the correct answers are returned at the first index, however, some correct answers are returned at the second and the third indices. The *Recall* value increases to 0.31 from 0.26 if the number of correct answers are based on the first three answers returned rather than the first answer returned.

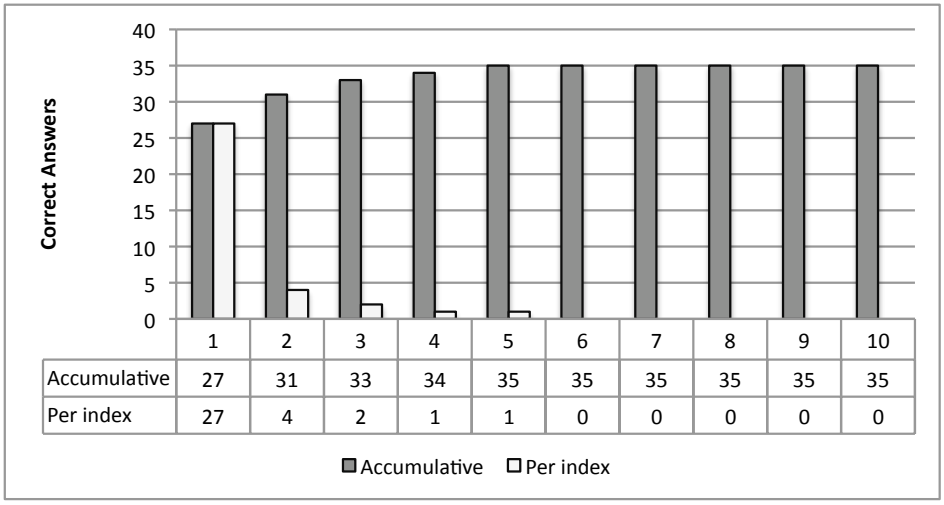


Figure 7.4: Correct answers returned by Stemmed String method

Figure 7.5 shows the effect of stemming. Here *Raw* means Raw String method, and *Stemmed* means Stemmed String method. A small increase occurs in the *MRR* and *Recall* values, and there is no change in the *Precision* value according to

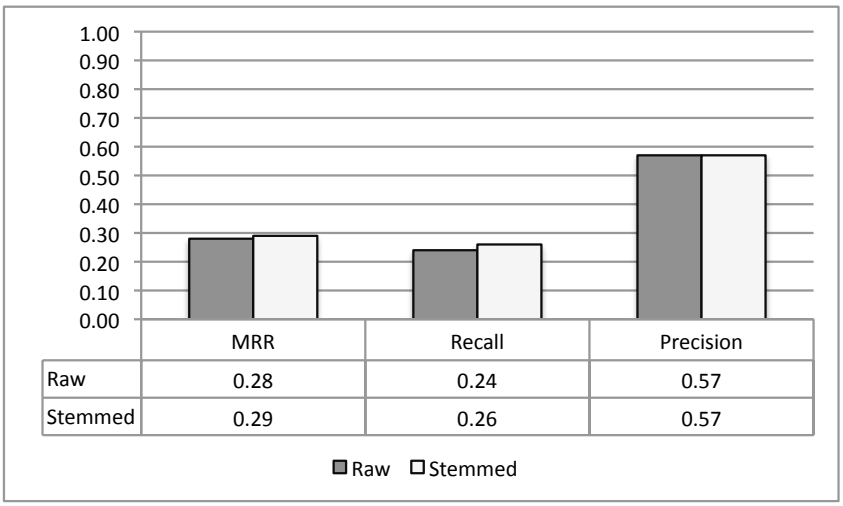


Figure 7.5: Effect of stemming

Figure 7.5. The aim of Stemmed String method is ignoring the affixes especially of the question phrases. We use many question-answer pairs to learn answer patterns and this yields to learn different forms (different affixes) of the question phrases. For instance, the following answer patterns are learned for *Capital* question type by Raw String method: "<Q>'nin başkenti <A>", "<Q>'nın başkenti <A>", "<Q>'in başkenti <A>", "<Q>'ın başkenti <A>", etc. The following answer pattern is learned by Stemmed String method: "<Q> başk <A>" and this answer pattern can extract an answer from the sentences which the answer patterns of Raw String method extract an answer. If we use less question-answer pairs for learning and "<Q>'ın başkenti <A>" answer pattern is not learned, the Raw String method extracts no answer for the question "Hırvatistan" from the following sentence, but Stemmed String method can extract the correct answer "Zagrep'tir".

"Hırvatistan'ın başkenti Zagrep'tir."

7.2.4 Method 4: Stemmed String with Answer Type

We present the results of Stemmed String with Answer Type method in this section. The system uses only the answer patterns learned by Stemmed String

with Answer Type method for this evaluation. The results are shown in Table 7.4. In Table 7.4, the results of *Capital* question type are the best results and the results of *DateOfBirth* question type are the worst results.

Question Type	#Q	#A	#CA	MRR	Precision	Recall	#AP
Author	15	5	5	0.33	1.00	0.33	140
Capital	15	12	12	0.80	1.00	0.80	776
DateOfBirth	15	0	0	0.00	0.00	0.00	26
DateOfDeath	15	1	1	0.07	1.00	0.07	154
Language	15	13	9	0.69	0.69	0.60	551
PlaceOfBirth	15	1	1	0.07	1.00	0.07	209
PlaceOfDeath	15	2	2	0.13	1.00	0.13	285
TOTAL	105	34	30	0.30	0.88	0.29	2141

Table 7.4: Results of Stemmed String with Answer Type method

Figure 7.6 shows the number of correct answers returned between the indices 1 - 10. According to Figure 7.6, most of the correct answers are returned at the first index, two correct answers are returned at the second index. There is only a small increase in the *Recall* value if the number of correct answers are based on the first three answers returned rather than the first answer returned.

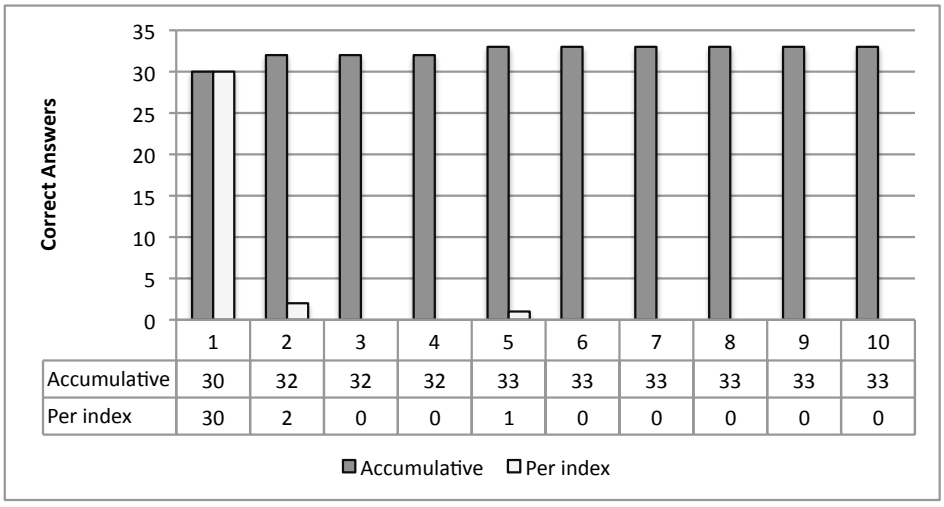


Figure 7.6: Correct answers returned by Stemmed String with Answer Type method

Figure 7.7 shows the effect of answer type checking for Stemmed String methods. Here *Stemmed* means Stemmed String method, and *StemmedNE* means

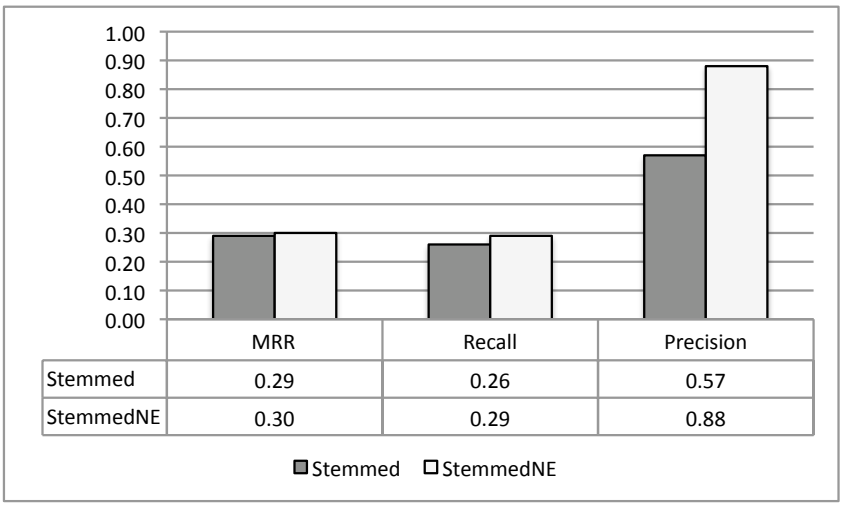


Figure 7.7: Effect of answer type checking for Stemmed String methods

Stemmed String with Answer Type method. *MRR*, *Recall* and *Precision* values of Stemmed String with Answer Type method are greater than *MRR*, *Recall* and *Precision* values of Stemmed String method as shown in Figure 7.7. Especially, a significant increase occurs in the *Precision* value. So, checking the answer type of a candidate answer before returning it as an answer increases the performance of the system.

7.2.5 Method 5: Named Entity Tagged String

We present the results of Named Entity Tagged String method in this section. The system uses only the answer patterns learned by Named Entity Tagged String method for this evaluation. The results are shown in Table 7.5.

According to Table 7.5, the best results are obtained for *Capital* question type and the worst results are obtained for *Language* question type. The reason is that the number of answer patterns learned for *Capital* question type is much more than the number of answer patterns learned for *Language* question type. For *Language* question type, Named Entity Tagged String method cannot learn any answer pattern whose confidence factor is more than 0.75 threshold.

Figure 7.8 shows the number of correct answers returned between the indices

Question Type	#Q	#A	#CA	MRR	Precision	Recall	#AP
Author	15	11	10	0.68	0.91	0.67	170
Capital	15	13	13	0.87	1.00	0.87	830
DateOfBirth	15	9	8	0.56	0.89	0.53	33
DateOfDeath	15	4	4	0.27	1.00	0.27	125
Language	15	0	0	0.00	0.00	0.00	0
PlaceOfBirth	15	10	10	0.67	1.00	0.67	205
PlaceOfDeath	15	3	2	0.13	0.67	0.13	278
TOTAL	105	50	47	0.45	0.94	0.45	1641

Table 7.5: Results of NE Tagged String method

1 - 10. According to Figure 7.8, most of the correct answers are returned at the first index, only one correct answer is returned at the third index. There is only a small increase in the *Recall* value if the number of correct answers are based on the first three answers returned rather than the first answer returned.

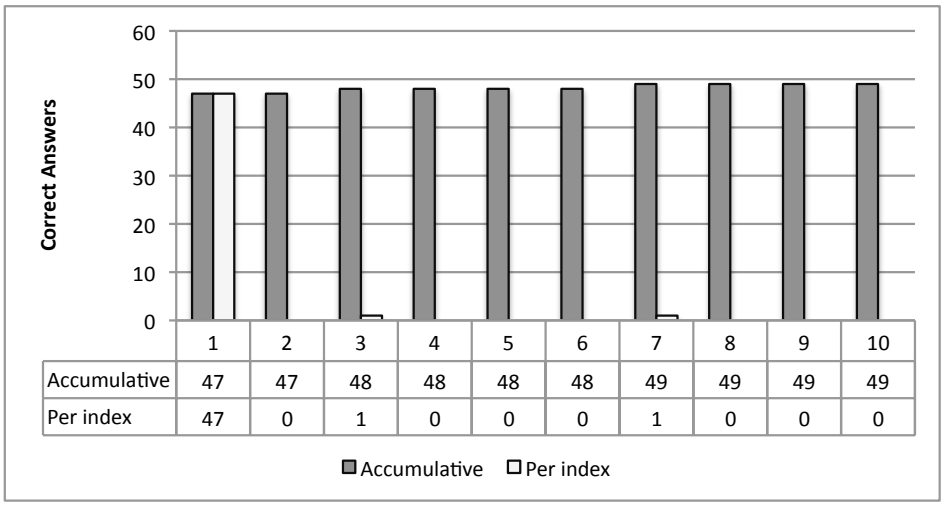
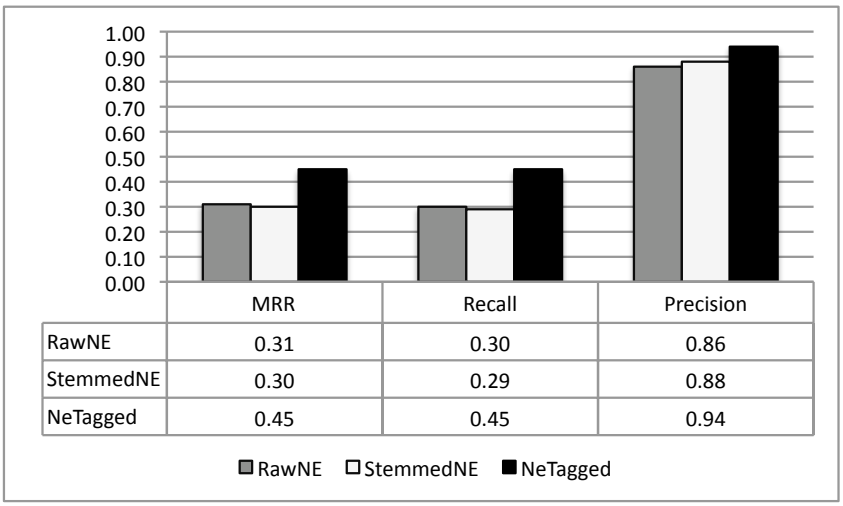


Figure 7.8: Correct answers returned by Named Entity Tagged String method

Figure 7.9 shows the comparison of the results of Named Entity Tagged String method and Raw and Stemmed String with Answer Type methods. Here *RawNE* means Raw String with Answer Type method, *StemmedNE* means Stemmed String with Answer Type method, and *NeTagged* means Named Entity Tagged String method. *MRR*, *Recall* and *Precision* values of Named Entity Tagged String method are greater than *MRR*, *Recall* and *Precision* values of the other



	MRR	Recall	Precision
RawNE	0.31	0.30	0.86
StemmedNE	0.30	0.29	0.88
NeTagged	0.45	0.45	0.94

Figure 7.9: Named Entity Tagged String method versus Raw and Stemmed String with Answer Type methods

two methods as shown in Figure 7.9.

7.2.6 Combining Methods without Answer Type

We give the results of combining methods without answer type in this section. The system uses answer patterns learned by Raw String, Stemmed String, and Named Entity Tagged String methods for this evaluation. The results are shown in Table 7.6. According to Table 7.6, the best results are obtained for *Capital* question type and the worst results are obtained for *PlaceOfDeath* and *DateOfDeath* question types.

Figure 7.10 shows the number of correct answers returned between the indices 1 - 10. According to Figure 7.10, most of the correct answers are returned at the first index, however, some correct answers are returned at the second and the third indices. The *Recall* value increases to 0.59 from 0.53 if the number of correct answers are based on the first three answers returned rather than the first answer returned.

Figure 7.11 shows the comparison of the results of combining methods without answer type and Raw, Stemmed, and Named Entity Tagged String methods.

Question Type	#Q	#A	#CA	MRR	Precision	Recall	#AP
Author	15	15	10	0.71	0.67	0.67	551
Capital	15	15	14	0.94	0.93	0.93	2958
DateOfBirth	15	9	8	0.56	0.89	0.53	89
DateOfDeath	15	9	3	0.22	0.33	0.20	462
Language	15	14	9	0.66	0.64	0.60	1148
PlaceOfBirth	15	15	10	0.68	0.67	0.67	847
PlaceOfDeath	15	4	2	0.17	0.50	0.13	1133
TOTAL	105	81	56	0.56	0.69	0.53	7188

Table 7.6: Results of combining methods without answer type

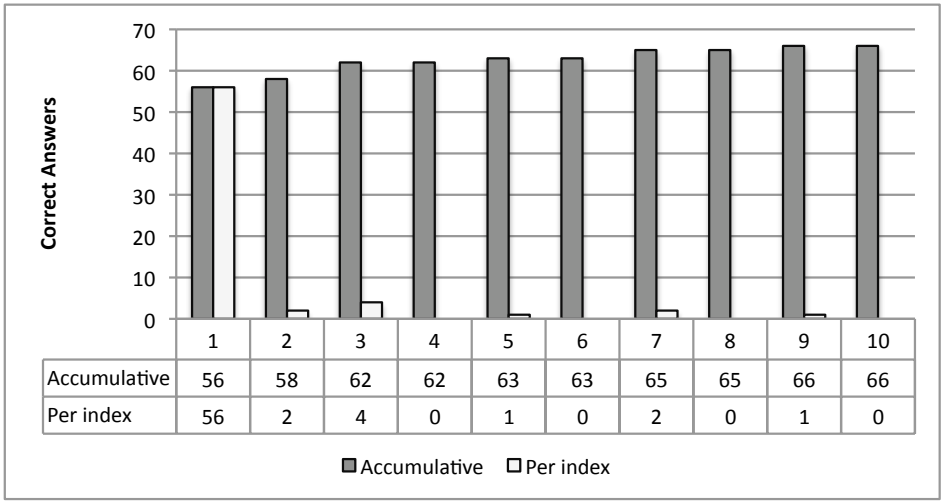


Figure 7.10: Correct answers returned by combining methods without answer type

Here *Raw* means Raw String method, *Stemmed* means Stemmed String method, *NeTagged* means Named Entity Tagged String method, and *AllwithoutNE* means combining methods without answer type. *MRR* and *Recall* values of combining methods without answer type are greater than *MRR* and *Recall* values of each individual method. *Precision* value of combining methods without answer type is greater than the *Precision* values of Raw and Stemmed String methods, but it is less than *Precision* value of Named Entity Tagged String method.

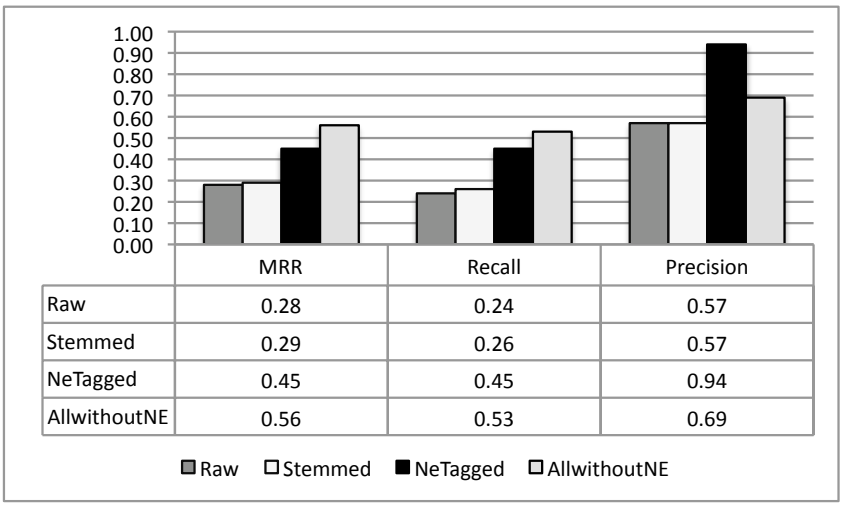


Figure 7.11: Comparison of the results of combining methods without answer type

7.2.7 Combining Methods with Answer Type

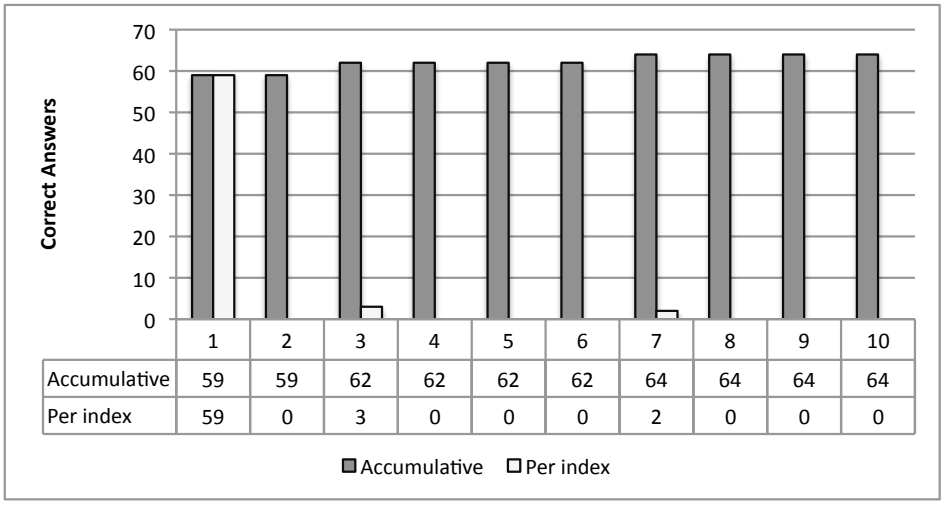
We give the results of combining methods with answer type in this section. The system uses answer patterns learned by Raw String with Answer Type, Stemmed String with Answer Type, and Named Entity Tagged String methods for this evaluation. The results are shown in Table 7.7. According to Table 7.7, the best results are obtained for *Capital* question type and the worst results are obtained for *PlaceOfDeath* question type.

Question Type	#Q	#A	#CA	MRR	Precision	Recall	#AP
Author	15	11	10	0.68	0.91	0.67	551
Capital	15	14	14	0.93	1.00	0.93	2525
DateOfBirth	15	9	8	0.56	0.89	0.53	89
DateOfDeath	15	6	4	0.27	0.67	0.27	462
Language	15	14	9	0.66	0.64	0.60	1148
PlaceOfBirth	15	11	11	0.73	1.00	0.73	687
PlaceOfDeath	15	4	3	0.20	0.75	0.20	890
TOTAL	105	69	59	0.58	0.86	0.56	6352

Table 7.7: Results of combining methods with answer type

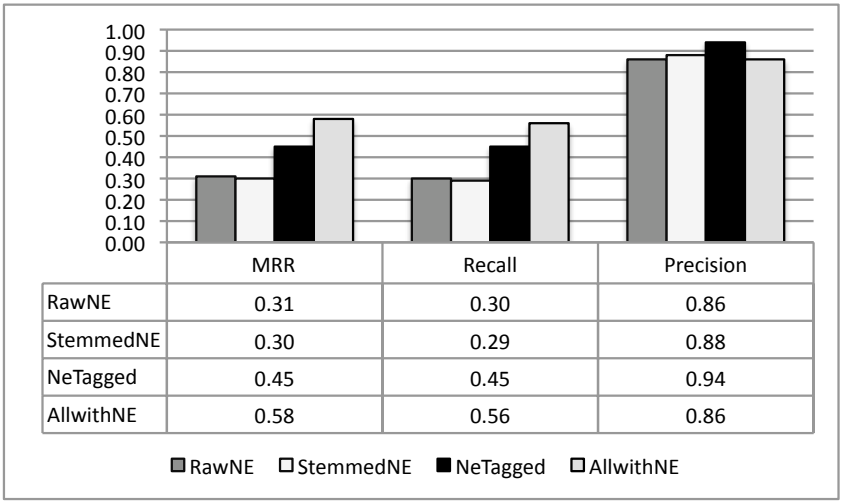
Figure 7.12 shows the number of correct answers returned between the indices 1 - 10. According to Figure 7.12, most of the correct answers are returned at the

first index, however, three correct answers are returned at the third index. The *Recall* value increases to 0.59 from 0.56 if the number of correct answers are based on the first three answers returned rather than the first answer returned.



	1	2	3	4	5	6	7	8	9	10
Accumulative	59	59	62	62	62	62	64	64	64	64
Per index	59	0	3	0	0	0	2	0	0	0

Figure 7.12: Correct answers returned by combining methods with answer type



	MRR	Recall	Precision
RawNE	0.31	0.30	0.86
StemmedNE	0.30	0.29	0.88
NeTagged	0.45	0.45	0.94
AllwithNE	0.58	0.56	0.86

Figure 7.13: Comparison of the results of combining methods with answer type

Figure 7.13 shows the comparison of the results of combining methods with answer type and Raw String with Answer Type, Stemmed String with Answer Type, and Named Entity Tagged String methods. Here *RawNE* means Raw String with Answer Type method, *StemmedNE* means Stemmed String with Answer Type method, *NeTagged* means Named Entity Tagged String method, and

AllwithNE means combining methods with answer type. *MRR* and *Recall* values of combining methods with answer type are greater than *MRR* and *Recall* values of each individual method. *Precision* value of combining methods with answer type is less than or equal to the *Precision* value of each individual method.

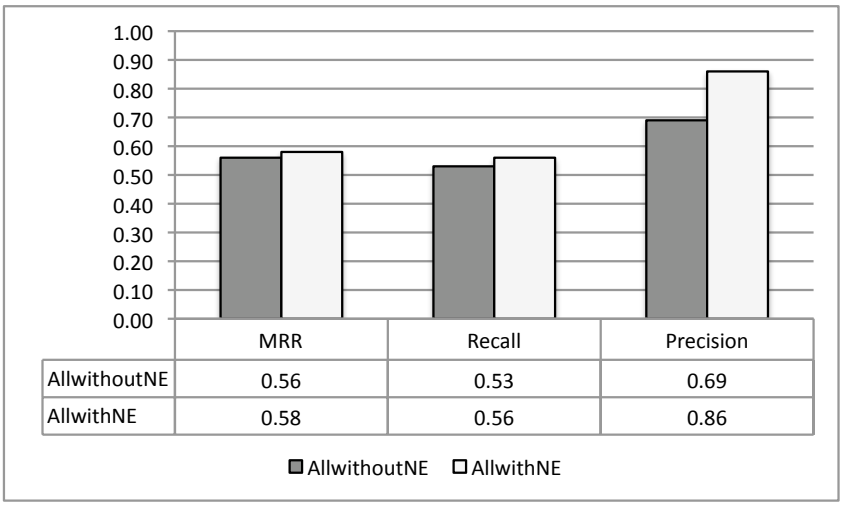


Figure 7.14: Comparison of the results of combining methods with answer type and without answer type

We compare the results obtained from combining methods with answer type and combining methods without answer type. The results are shown in Figure 7.14. *MRR*, *Recall*, and Precision values of combining methods with answer type are greater than *MRR*, *Recall*, and Precision values of combining methods without answer type.

7.2.8 Effect of Confidence Factor Threshold

In this section, we discuss the effect of the selected confidence factor threshold on the performance of the system. First, the effect of confidence factor threshold on the *MRR* and *Recall* values is evaluated. Then, the effect of confidence factor threshold on the *Precision* and *Recall* values along with *F-measure* values is evaluated.

A figure is given for each method to show the *MRR* and *Recall* values at

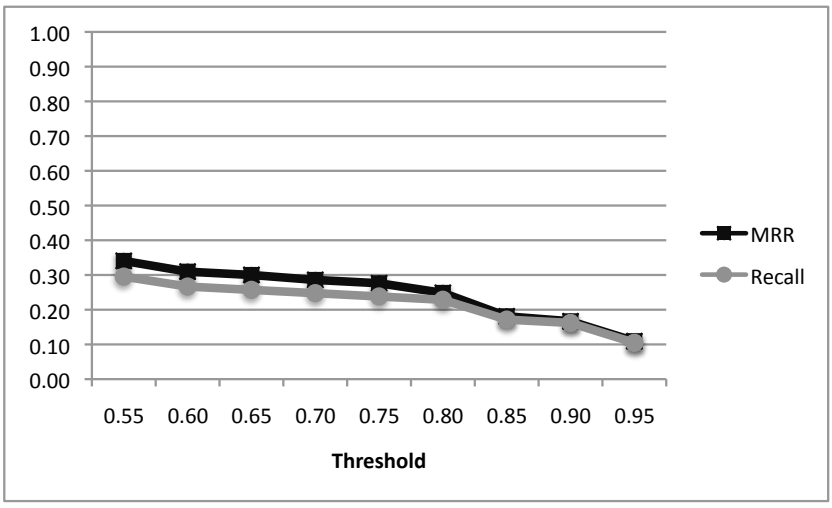


Figure 7.15: *MRR* and *Recall* values of Raw String method at different confidence factor thresholds

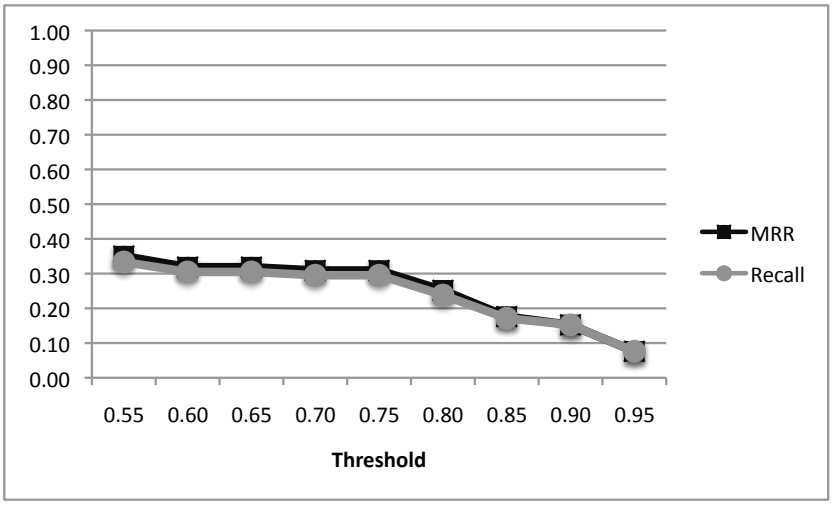


Figure 7.16: *MRR* and *Recall* values of Raw String with Answer Type method at different confidence factor thresholds

different confidence factor thresholds:

- Figure 7.15 shows the *MRR* and *Recall* values for Raw String method at different confidence factor thresholds.
- Figure 7.16 shows the *MRR* and *Recall* values for Raw String with Answer Type method at different confidence factor thresholds.
- Figure 7.17 shows the *MRR* and *Recall* values for Stemmed String method

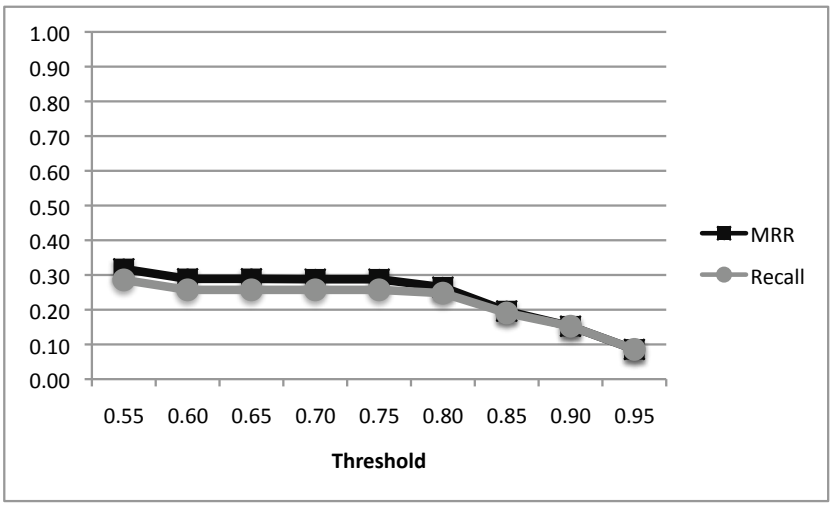


Figure 7.17: *MRR* and *Recall* values of Stemmed String method at different confidence factor thresholds

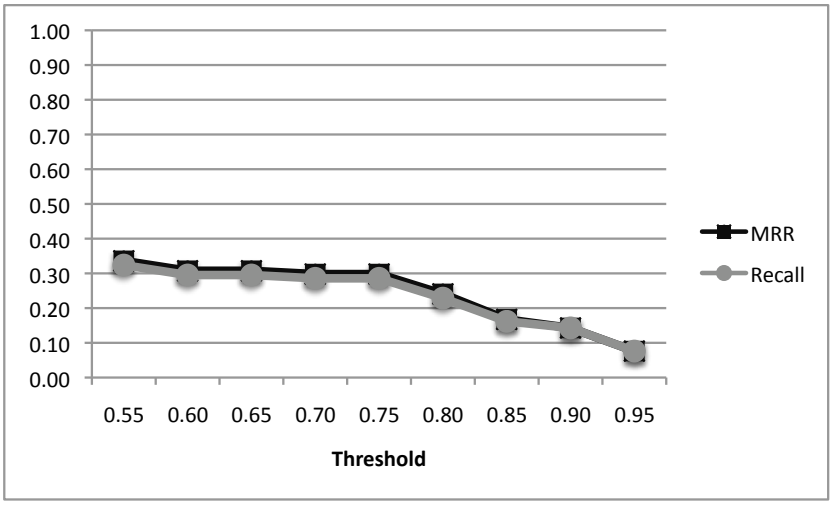


Figure 7.18: *MRR* and *Recall* values of Stemmed String with Answer Type method at different confidence factor thresholds

at different confidence factor thresholds.

- Figure 7.18 shows the *MRR* and *Recall* values for Stemmed String with Answer Type method at different confidence factor thresholds.
- Figure 7.19 shows the *MRR* and *Recall* values for Named Entity Tagged String method at different confidence factor thresholds.

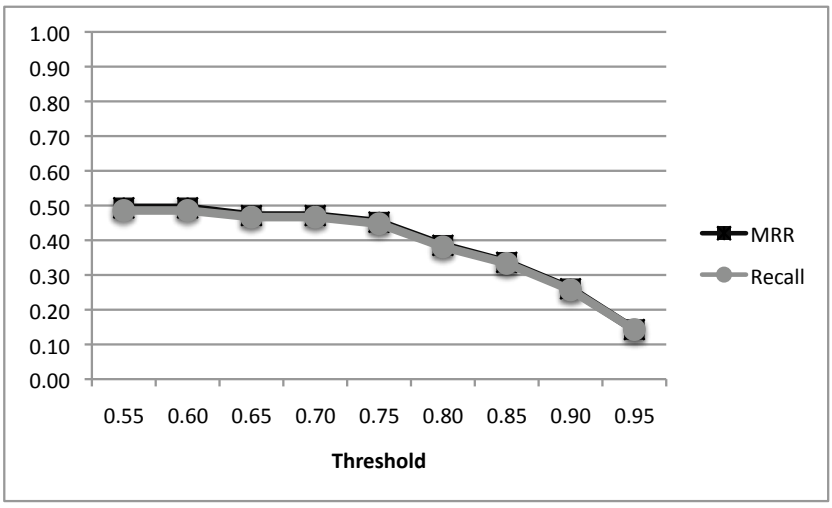


Figure 7.19: *MRR* and *Recall* values of Named Entity Tagged String method at different confidence factor thresholds

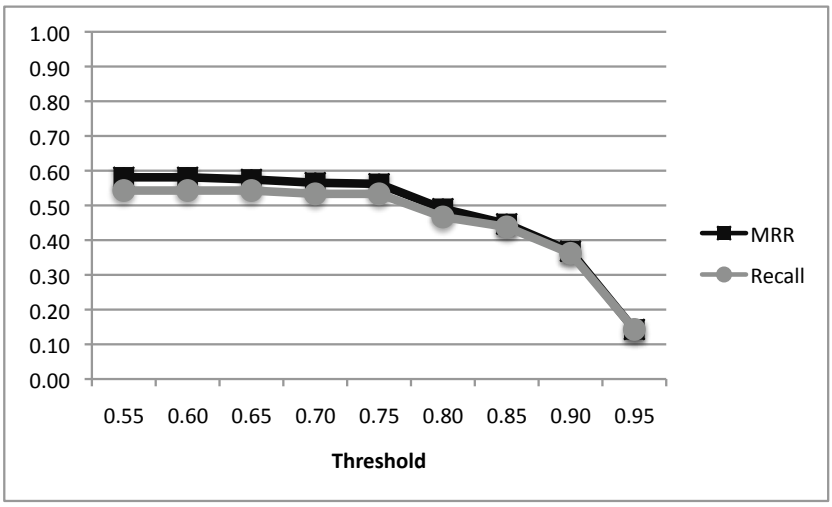


Figure 7.20: *MRR* and *Recall* values of combining methods without answer type at different confidence factor thresholds

- Figure 7.20 shows the *MRR* and *Recall* values for combining methods without answer type at different confidence factor thresholds.
- Figure 7.21 shows the *MRR* and *Recall* values for combining methods with answer type at different confidence factor thresholds.

According to figures, *MRR* and *Recall* graphics are similar. The similarity shows that correct answers are generally returned at the first index. *MRR* and

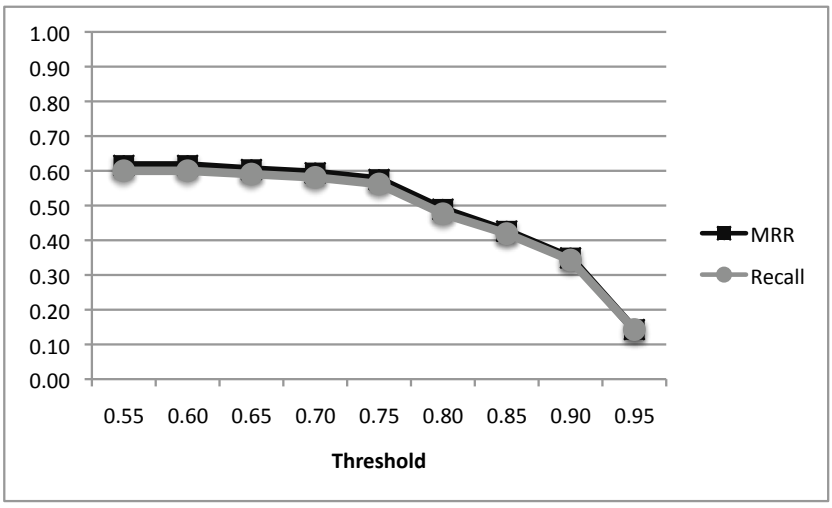


Figure 7.21: *MRR* and *Recall* values of combining methods with answer type at different confidence factor thresholds

Recall graphics are significantly similar for the methods which check answer type. These methods do not return answers if the named entity tag of the candidate answer and the expected type of the answer are not the same. Checking answer type decreases the number of incorrect answers.

A figure is given for each method to show the *Precision* and *Recall* values along with *F-measure* values at different confidence factor thresholds:

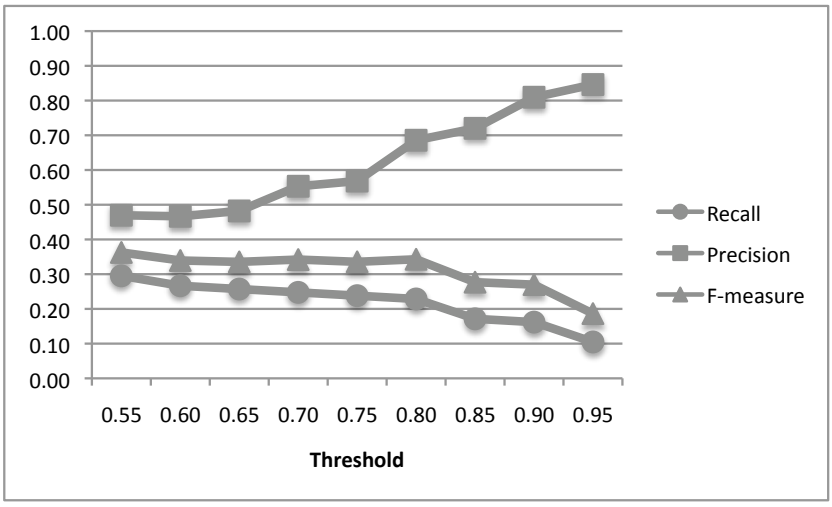


Figure 7.22: *Precision* and *Recall* values of Raw String method at different confidence factor thresholds

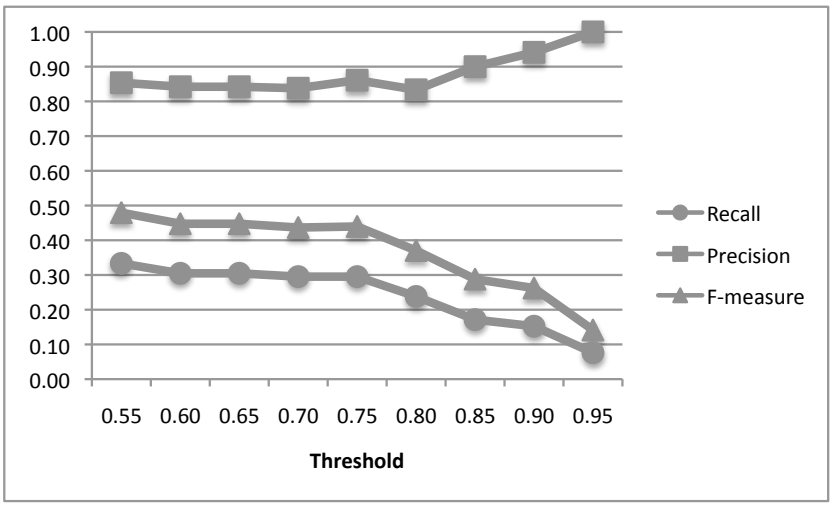


Figure 7.23: *Precision* and *Recall* values of Raw String with Answer Type method at different confidence factor thresholds

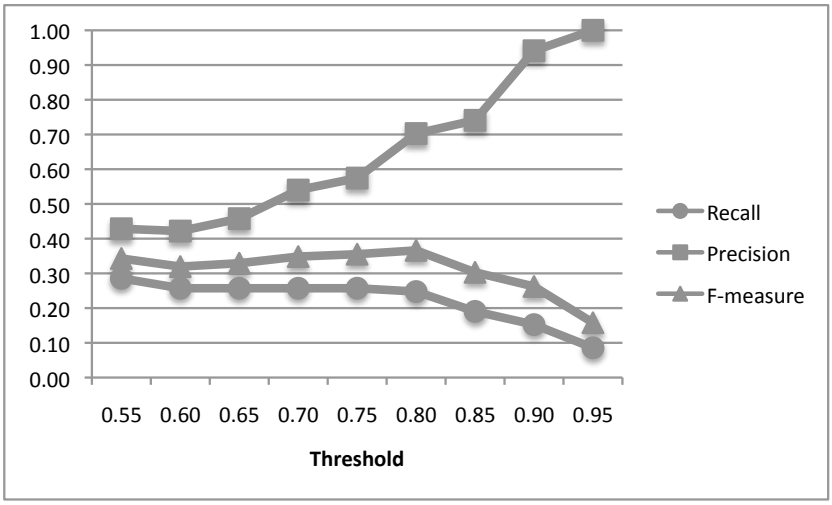


Figure 7.24: *Precision* and *Recall* values of Stemmed String method at different confidence factor thresholds

- Figure 7.22 shows the *Precision* and *Recall* values along with *F-measure* values for Raw String method at different confidence factor thresholds.
- Figure 7.23 shows the *Precision* and *Recall* values along with *F-measure* values for Raw String with Answer Type method at different confidence factor thresholds.
- Figure 7.24 shows the *Precision* and *Recall* values along with *F-measure*

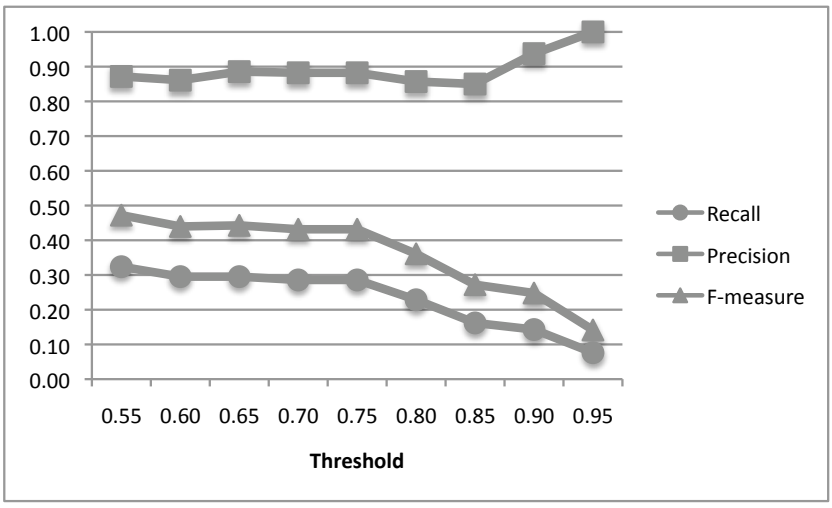


Figure 7.25: *Precision* and *Recall* values of Stemmed String with Answer Type method at different confidence factor thresholds

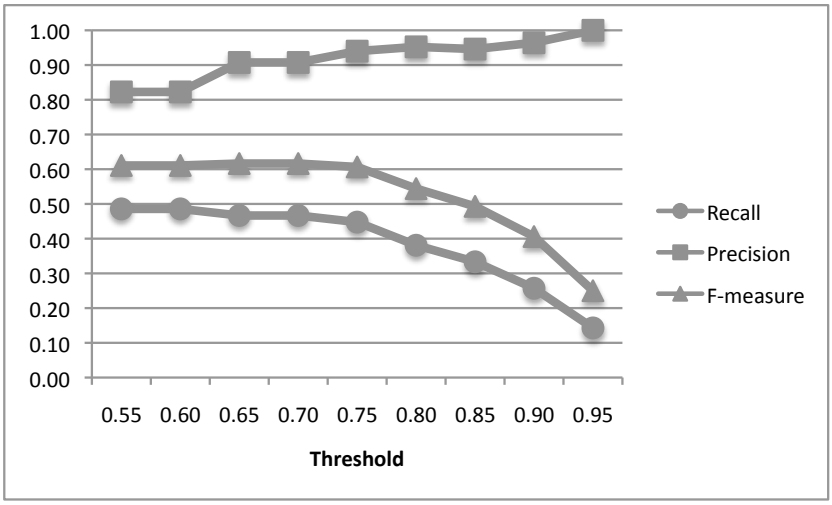


Figure 7.26: *Precision* and *Recall* values of Named Entity Tagged String method at different confidence factor thresholds

values for Stemmed String method at different confidence factor thresholds.

- Figure 7.25 shows the *Precision* and *Recall* values along with *F-measure* values for Stemmed String with Answer Type method at different confidence factor thresholds.

- Figure 7.26 shows the *Precision* and *Recall* values along with *F-measure* values for Named Entity Tagged String method at different confidence factor

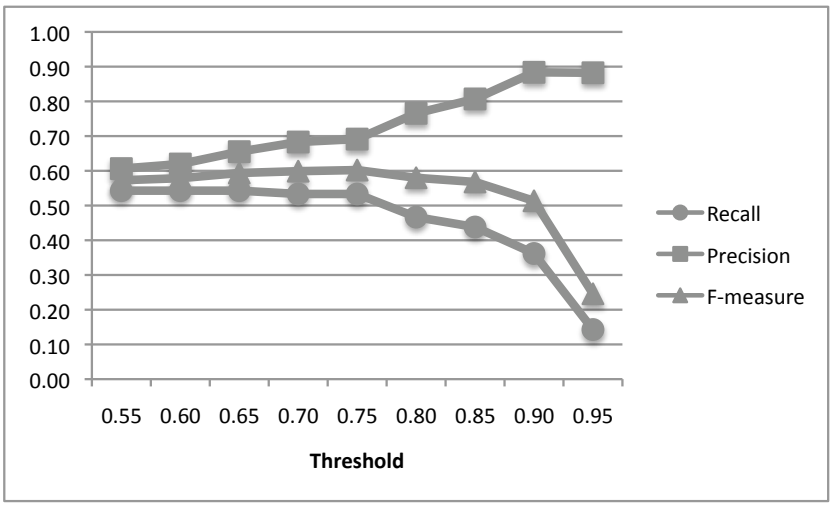


Figure 7.27: *Precision* and *Recall* values of combining methods without answer type at different confidence factor thresholds

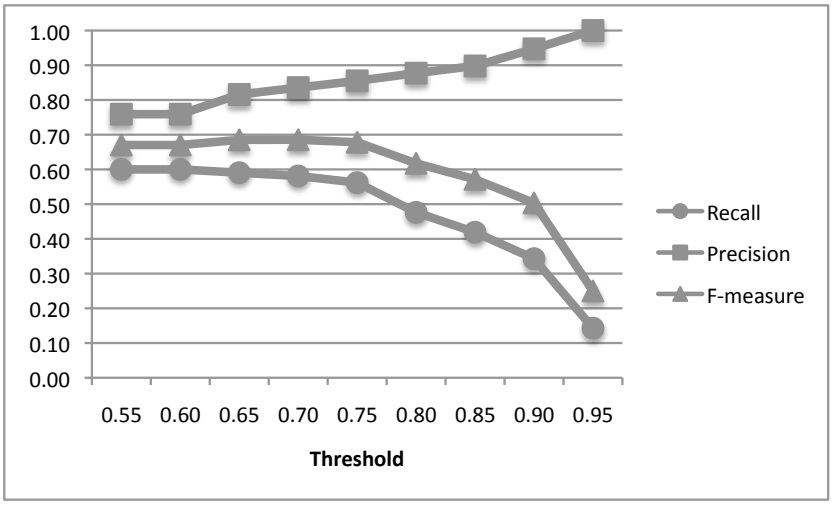


Figure 7.28: *Precision* and *Recall* values of combining methods with answer type at different confidence factor thresholds

thresholds.

- Figure 7.27 shows the *Precision* and *Recall* values along with *F-measure* values for combining methods without answer type at different confidence factor thresholds.

- Figure 7.28 shows the *Precision* and *Recall* values along with *F-measure* values for combining methods with answer type at different confidence factor

thresholds.

According to figures, while the confidence factor increases, *Precision* increases and *Recall* decreases. Using the confidence factor which both *Precision* and *Recall* values are maximized yields a better performance. *F-measure* is used to find the maximization point. 0.75 can be selected as confidence factor threshold for all methods.

7.3 Evaluation of Answer Re-ranking

We evaluate the answer re-ranking approach in this section. Answer re-ranking approach is based on frequency counting as explained in Chapter 6. The results of answer re-ranking approach are given in the following sections for each method. The effect of answer re-ranking approach on the performance of the system is also discussed in the following sections. In the evaluations, confidence factor threshold is selected as 0.55.

7.3.1 Method 1: Raw String

We present the results of answer re-ranking approach for Raw String method in this section. The system uses only the answer patterns learned by Raw String method for this evaluation. The results are shown in Table 7.8. According to Table 7.8, the best results are obtained for *Capital* question type and the worst results are obtained for *PlaceOfDeath* question type.

Figure 7.29 shows the number of correct answers returned between the indices 1 - 10. According to Figure 7.29, most of the correct answers are returned at the first index, however, some correct answers are returned at the second and the third indices. The *Recall* value increases to 0.44 from 0.38 if the number of correct answers are based on the first three answers returned rather than the first answer returned.

Question Type	#Q	#A	#CA	MRR	Precision	Recall	#AP
Author	15	14	5	0.39	0.36	0.33	360
Capital	15	15	13	0.93	0.87	0.87	3288
DateOfBirth	15	7	3	0.20	0.43	0.20	90
DateOfDeath	15	9	2	0.13	0.22	0.13	390
Language	15	15	11	0.73	0.73	0.73	2164
PlaceOfBirth	15	14	5	0.39	0.36	0.33	716
PlaceOfDeath	15	5	1	0.07	0.20	0.07	861
TOTAL	105	79	40	0.41	0.51	0.38	7869

Table 7.8: Results of Raw String method with answer re-ranking

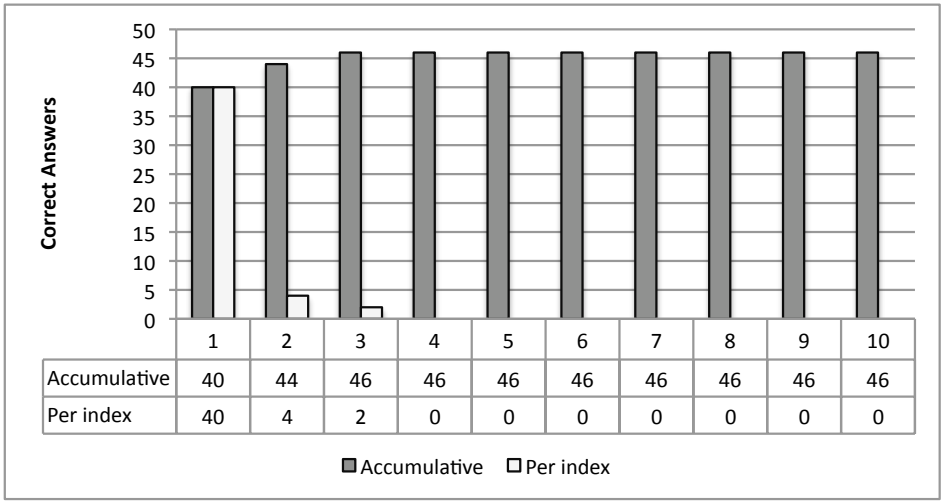


Figure 7.29: Correct answers returned by Raw String method with answer re-ranking

Figure 7.30 shows the results of Raw String method when answer re-ranking is not applied and the results of Raw String method when answer re-ranking is applied. Answer re-ranking improves the *MRR*, *Recall*, and *Precision* values according to Figure 7.30.

7.3.2 Method 2: Raw String with Answer Type

We give the results of answer re-ranking approach for Raw String with Answer Type method in this section. The system uses only the answer patterns learned by

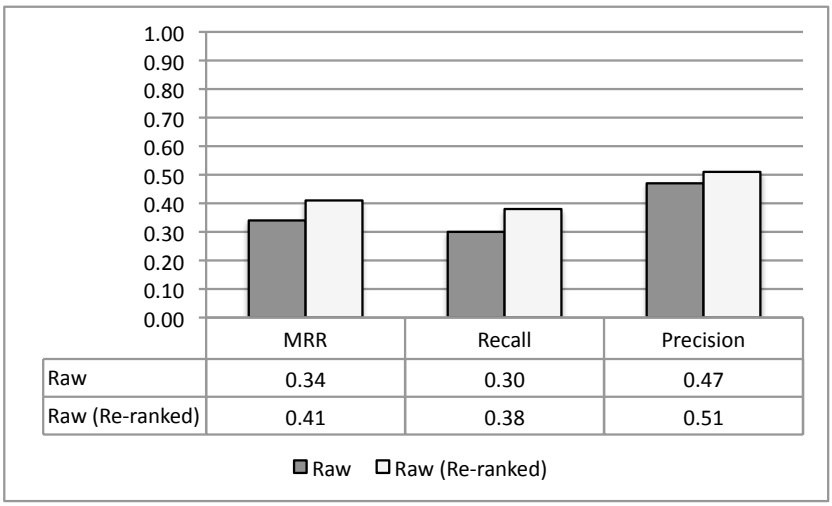


Figure 7.30: Comparison of the results of answer re-ranking for Raw String method

Raw String with Answer Type method for this evaluation. The results are shown in Table 7.9. According to Table 7.9, the best results are obtained for *Capital* question type and the worst results are obtained for *PlaceOfDeath* question type.

Question Type	#Q	#A	#CA	MRR	Precision	Recall	#AP
Author	15	7	7	0.47	1.00	0.47	360
Capital	15	14	14	0.93	1.00	0.93	2654
DateOfBirth	15	3	3	0.20	1.00	0.20	90
DateOfDeath	15	5	2	0.13	0.40	0.13	390
Language	15	15	11	0.73	0.73	0.73	2164
PlaceOfBirth	15	4	3	0.20	0.75	0.20	516
PlaceOfDeath	15	1	1	0.07	1.00	0.07	617
TOTAL	105	49	41	0.39	0.84	0.39	6791

Table 7.9: Results of Raw String with Answer Type method with answer re-ranking

Figure 7.31 shows the number of correct answers returned between the indices 1 - 10. According to Figure 7.31, all of the correct answers are returned at the first index.

Figure 7.32 shows the results of Raw String with Answer Type method when answer re-ranking is not applied and the results of Raw String with Answer Type

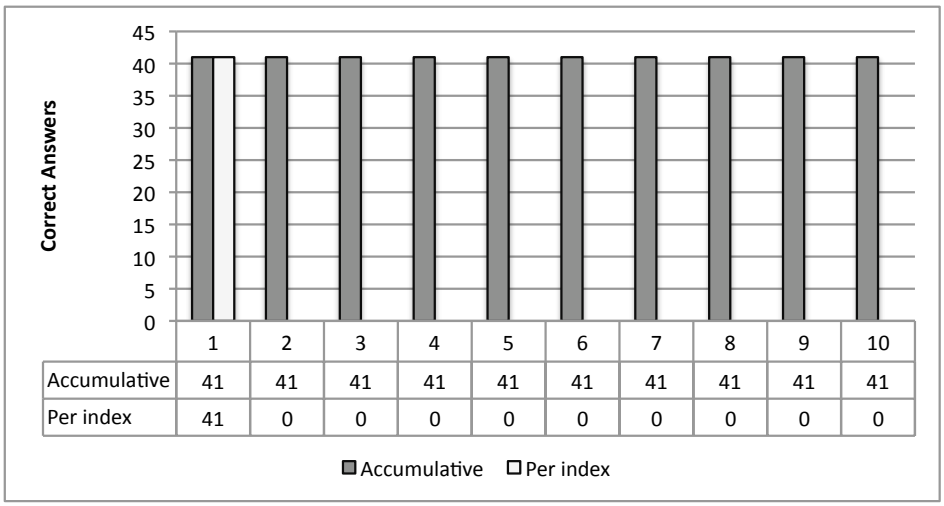


Figure 7.31: Correct answers returned by Raw String with Answer Type method with answer re-ranking

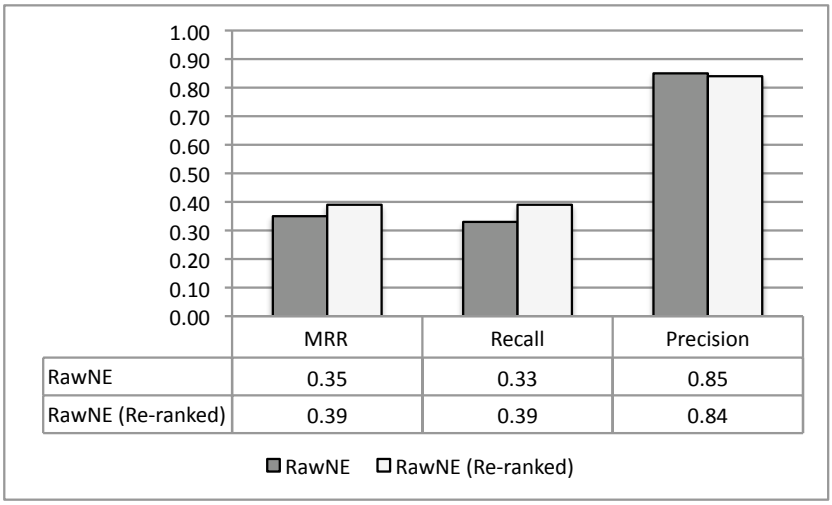


Figure 7.32: Comparison of the results of answer re-ranking for Raw String with Answer Type method

method when answer re-ranking is applied. Answer re-ranking improves the *MRR* and *Recall*, but it degrades the *Precision*.

7.3.3 Method 3: Stemmed String

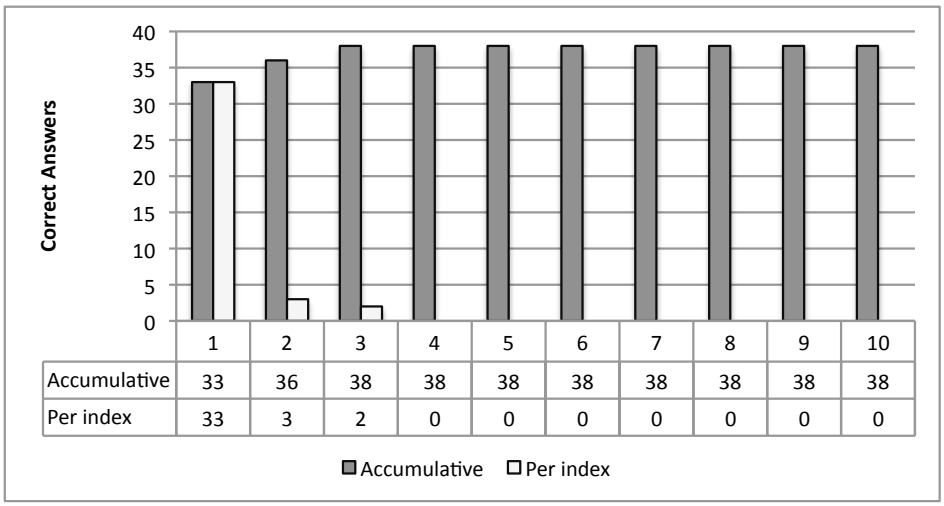
We give the results of answer re-ranking approach for Stemmed String method in this section. The system uses only the answer patterns learned by Stemmed String method for this evaluation. The results are shown in Table 7.10. According to Table 7.10, the best results are obtained for *Capital* question type and the worst results are obtained for *DateOfDeath* and *PlaceOfBirth* question types.

Question Type	#Q	#A	#CA	MRR	Precision	Recall	#AP
Author	15	14	3	0.26	0.21	0.20	196
Capital	15	15	14	0.97	0.93	0.93	2695
DateOfBirth	15	7	3	0.20	0.43	0.20	76
DateOfDeath	15	9	1	0.07	0.11	0.07	316
Language	15	15	9	0.66	0.60	0.60	2043
PlaceOfBirth	15	12	1	0.07	0.08	0.07	514
PlaceOfDeath	15	6	2	0.13	0.33	0.13	721
TOTAL	105	78	33	0.33	0.42	0.31	6561

Table 7.10: Results of Stemmed String method with answer re-ranking

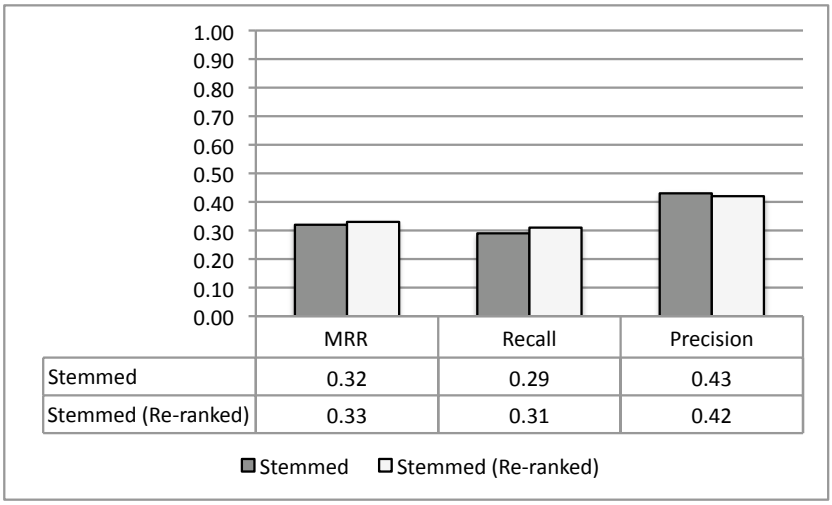
Figure 7.33 shows the number of correct answers returned between the indices 1 - 10. According to Figure 7.33, most of the correct answers are returned at the first index, however, some correct answers are returned at the second and the third indices. The *Recall* value increases to 0.36 from 0.31 if the number of correct answers are based on the first three answers returned rather than the first answer returned.

Figure 7.34 shows the results of Stemmed String method when answer re-ranking is not applied and the results of Stemmed String method when answer re-ranking is applied. *MRR* and *Recall* values increase when answer re-ranking is applied, however, *Precision* value decreases when answer re-ranking is applied.



	1	2	3	4	5	6	7	8	9	10
Accumulative	33	36	38	38	38	38	38	38	38	38
Per index	33	3	2	0	0	0	0	0	0	0

Figure 7.33: Correct answers returned by Stemmed String method with answer re-ranking



	MRR	Recall	Precision
Stemmed	0.32	0.29	0.43
Stemmed (Re-ranked)	0.33	0.31	0.42

Figure 7.34: Comparison of the results of answer re-ranking for Stemmed String method

7.3.4 Method 4: Stemmed String with Answer Type

We present the results of answer re-ranking approach for Stemmed String with Answer Type method in this section. The system uses only the answer patterns learned by Stemmed String with Answer Type method for this evaluation. The results are shown in Table 7.11. According to Table 7.11, the best results are

obtained for *Capital* question type and the worst results are obtained for *DateOfDeath* and *PlaceOfBirth* question types.

Question Type	#Q	#A	#CA	MRR	Precision	Recall	#AP
Author	15	6	5	0.33	0.83	0.33	196
Capital	15	14	14	0.93	1.00	0.93	2310
DateOfBirth	15	3	3	0.20	1.00	0.20	76
DateOfDeath	15	3	1	0.07	0.33	0.07	316
Language	15	15	9	0.66	0.60	0.60	2043
PlaceOfBirth	15	2	1	0.07	0.50	0.07	395
PlaceOfDeath	15	2	2	0.13	1.00	0.13	538
TOTAL	105	45	35	0.34	0.78	0.33	5874

Table 7.11: Results of Stemmed String with Answer Type method with answer re-ranking

Figure 7.35 shows the number of correct answers returned between the indices 1 - 10. According to Figure 7.35, most of the correct answers are returned at the first index, a few correct answers are returned at the second and the third indices. There is only a small increase in the *Recall* value if the number of correct answers are based on the first three answers returned rather than the first answer returned.

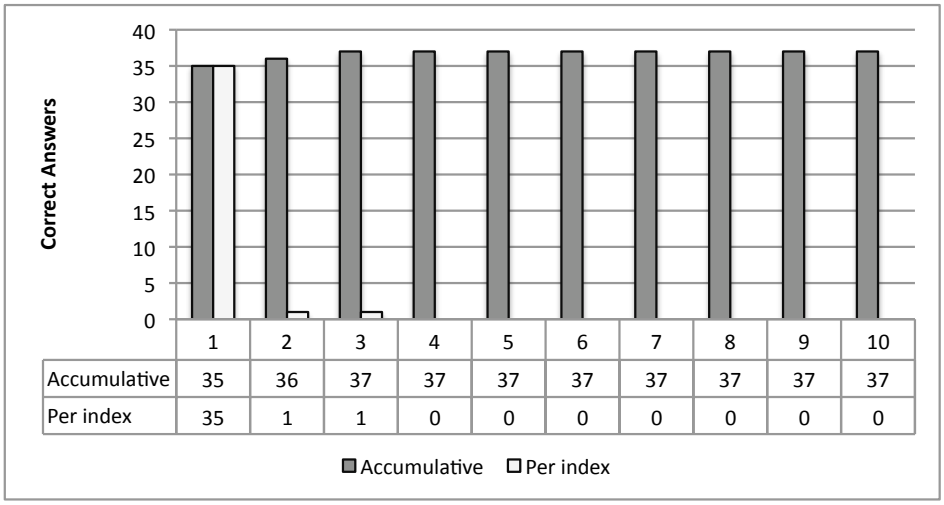


Figure 7.35: Correct answers returned by Stemmed String with Answer Type method with answer re-ranking

Figure 7.36 shows the results of Stemmed String with Answer Type method when answer re-ranking is not applied and the results of Stemmed String with

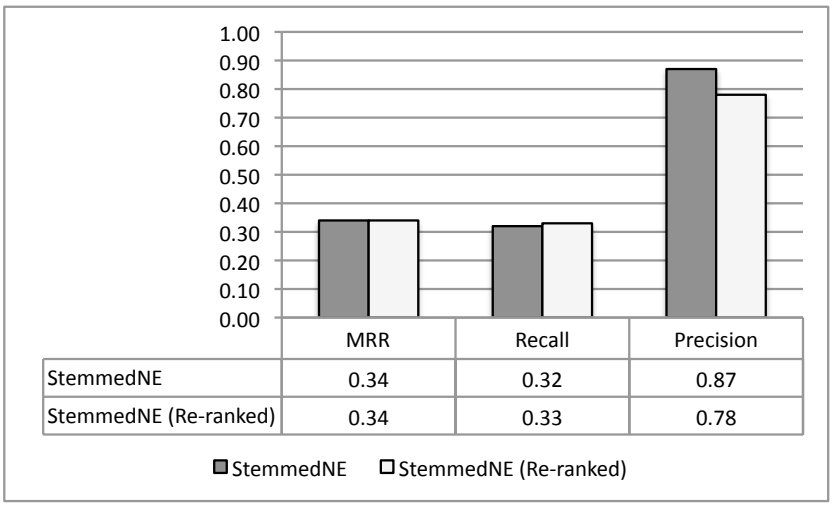


Figure 7.36: Comparison of the results of answer re-ranking for Stemmed String with Answer Type method

Answer Type method when answer re-ranking is applied. There is no change in the *MRR* value when answer re-ranking is applied. Answer re-ranking improves the *Recall*, but it degrades *Precision*.

7.3.5 Method 5: Named Entity Tagged String

We give the results of answer re-ranking approach for Named Entity Tagged String method in this section. The system uses only the answer patterns learned by Named Entity Tagged String method for this evaluation. The results are shown in Table 7.12. According to Table 7.12, the best results are obtained for *Capital* question type and the worst results are obtained for *Language* question type.

Figure 7.37 shows the number of correct answers returned between the indices 1 - 10. According to Figure 7.35, most of the correct answers are returned at the first index, two correct answers are returned at the second index. There is only a small increase in the *Recall* value if the number of correct answers are based on the first three answers returned rather than the first answer returned.

Figure 7.38 shows the results of Named Entity Tagged String method when

Question Type	#Q	#A	#CA	MRR	Precision	Recall	#AP
Author	15	14	9	0.65	0.64	0.60	240
Capital	15	14	14	0.93	1.00	0.93	2375
DateOfBirth	15	11	9	0.60	0.82	0.60	74
DateOfDeath	15	5	5	0.33	1.00	0.33	235
Language	15	0	0	0.00	0.00	0.00	0
PlaceOfBirth	15	14	12	0.83	0.86	0.80	410
PlaceOfDeath	15	7	2	0.13	0.29	0.13	522
TOTAL	105	65	51	0.50	0.78	0.49	3856

Table 7.12: Results of NE Tagged String method with answer re-ranking

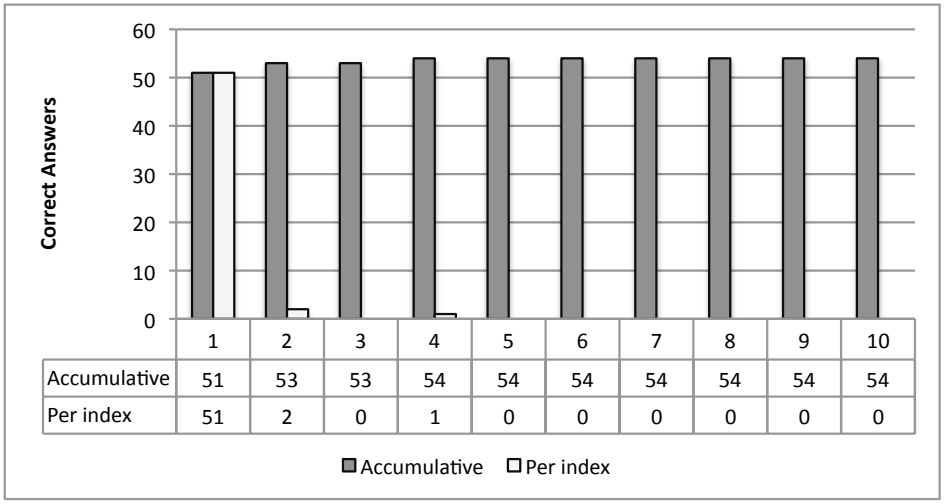


Figure 7.37: Correct answers returned by Named Entity Tagged String method with answer re-ranking

answer re-ranking is not applied and the results of Named Entity Tagged String method when answer re-ranking is applied. There is no change in the *MRR* and *Recall* values when answer re-ranking is applied. *Precision* value decreases when answer re-ranking is applied.

7.3.6 Combining Methods without Answer Type

We present the results of answer re-ranking approach for combining methods without answer type in this section. The system uses the answer patterns learned

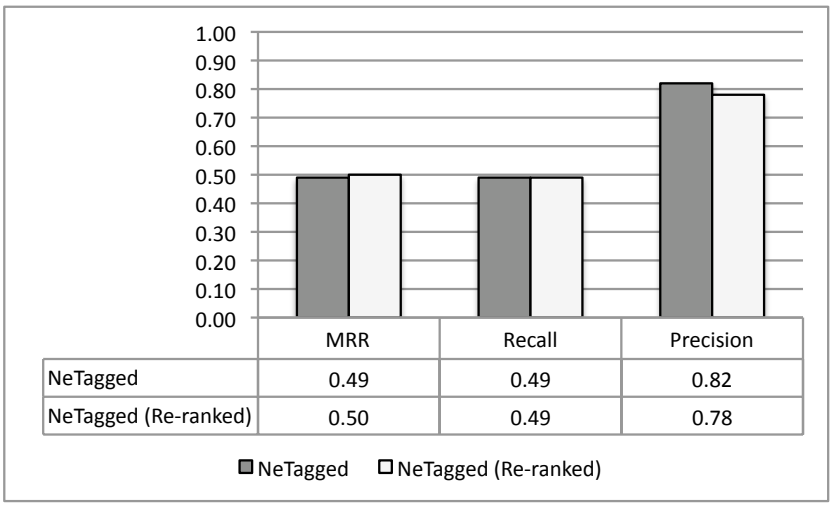


Figure 7.38: Comparison of the results of answer re-ranking for Named Entity Tagged String method

by Raw String, Stemmed String, and Named Entity Tagged String methods for this evaluation. The results are shown in Table 7.13. According to Table 7.13, the best results are obtained for *Capital* question type and the worst results are obtained for *PlaceOfDeath* question type.

Question Type	#Q	#A	#CA	MRR	Precision	Recall	#AP
Author	15	15	9	0.65	0.60	0.60	796
Capital	15	15	14	0.97	0.93	0.93	8358
DateOfBirth	15	13	9	0.60	0.69	0.60	240
DateOfDeath	15	13	6	0.40	0.46	0.40	941
Language	15	15	10	0.70	0.67	0.67	4207
PlaceOfBirth	15	15	9	0.73	0.60	0.60	1640
PlaceOfDeath	15	12	3	0.20	0.25	0.20	2104
TOTAL	105	98	60	0.61	0.61	0.57	18286

Table 7.13: Results of combining methods without answer type with answer re-ranking

Figure 7.39 shows the number of correct answers returned between the indices 1 - 10. According to Figure 7.39, most of the correct answers are returned at the first index, however, seven correct answers are returned at the second index. The *Recall* value increases to 0.64 from 0.57 if the number of correct answers are based on the first three answers returned rather than the first answer returned.

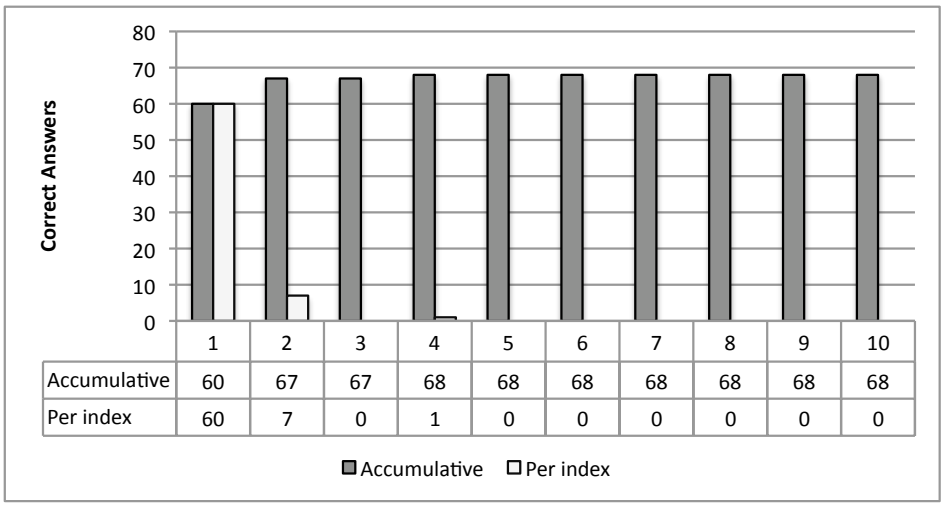


Figure 7.39: Correct answers returned by combining methods without answer type with answer re-ranking

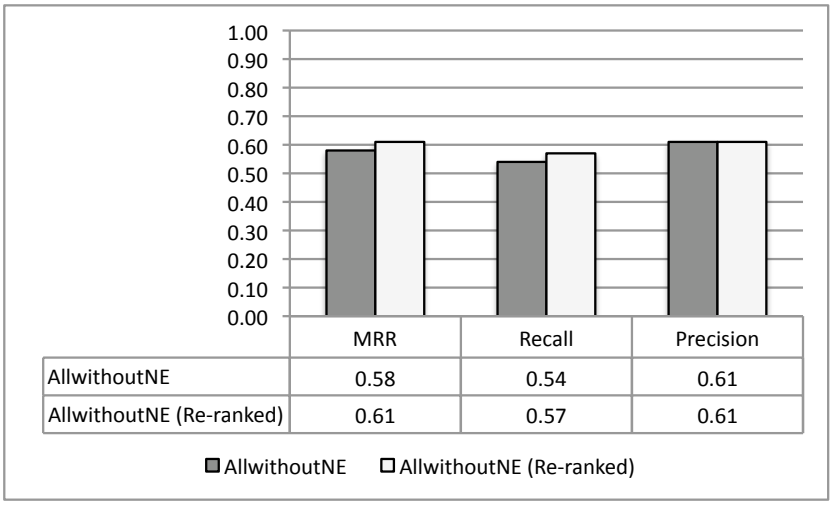


Figure 7.40: Comparison of the results of answer re-ranking for combining methods without answer type

Figure 7.40 shows the results of combining methods without answer type and the results of combining methods without answer type when answer re-ranking is applied. Both of the *MRR* and *Recall* values increase when answer re-ranking is applied, and there is no change in the *Precision* value when answer re-ranking is applied.

7.3.7 Combining Methods with Answer Type

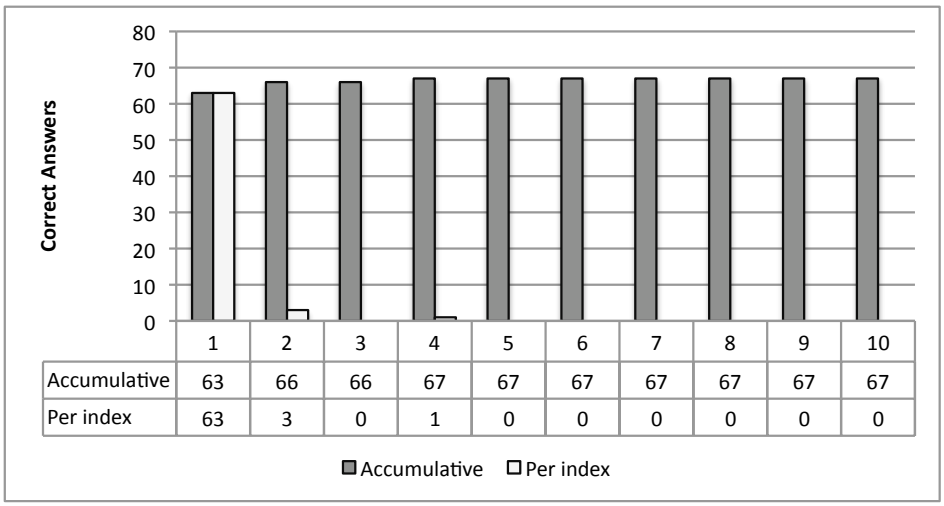
We present the results of answer re-ranking approach for combining methods with answer type in this section. The system uses the answer patterns learned by Raw String with Answer Type, Stemmed String with Answer Type, and Named Entity Tagged String methods for this evaluation. The results are shown in Table 7.14. According to Table 7.14, the best results are obtained for *Capital* question type and the worst results are obtained for *PlaceOfDeath* question type.

Question Type	#Q	#A	#CA	MRR	Precision	Recall	#AP
Author	15	14	9	0.65	0.64	0.60	796
Capital	15	14	14	0.93	1.00	0.93	7339
DateOfBirth	15	11	9	0.60	0.82	0.60	240
DateOfDeath	15	8	6	0.40	0.75	0.40	941
Language	15	15	10	0.70	0.67	0.67	4207
PlaceOfBirth	15	14	12	0.83	0.86	0.80	1321
PlaceOfDeath	15	8	3	0.20	0.38	0.20	1677
TOTAL	105	84	63	0.62	0.75	0.60	16521

Table 7.14: Results of combining methods with answer type with answer re-ranking

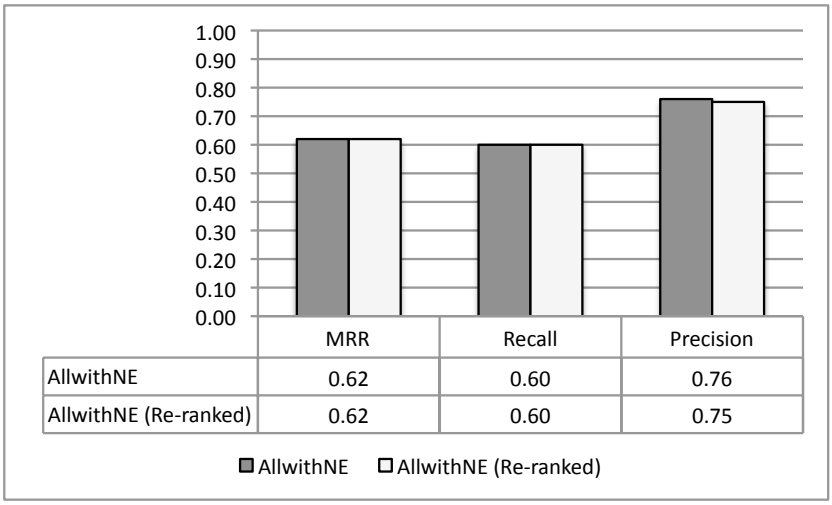
Figure 7.41 shows the number of correct answers returned between the indices 1 - 10. According to Figure 7.41, most of the correct answers are returned at the first index, however, three correct answers are returned at the second index. The *Recall* value increases to 0.63 from 0.60 if the number of correct answers are based on the first three answers returned rather than the first answer returned.

Figure 7.42 shows the results of combining methods with answer type and the results of combining methods with answer type when answer re-ranking is applied. There is no change in the *MRR* and *Recall* values when answer re-ranking is applied, and the *Precision* value decreases when answer re-ranking is applied.



	1	2	3	4	5	6	7	8	9	10
Accumulative	63	66	66	67	67	67	67	67	67	67
Per index	63	3	0	1	0	0	0	0	0	0

Figure 7.41: Correct answers returned by combining methods with answer type with answer re-ranking



	MRR	Recall	Precision
AllwithNE	0.62	0.60	0.76
AllwithNE (Re-ranked)	0.62	0.60	0.75

Figure 7.42: Comparison of the results of answer re-ranking for combining methods with answer type

7.4 Evaluation of Query Expansion

We evaluate our query expansion approach in this section. Our query expansion approach is explained in Chapter 6. We apply query expansion for Raw String and Raw String with Answer Type methods. Query expansion cannot be applied for Stemmed String and Named Entity Tagged String methods. The answer patterns whose confidence factor is equal to or greater than 0.75 are used for

query expansion.

7.4.1 Effect of Query Expansion on Document and Sentence Retrieval

The effect of query expansion on Document and Sentence Retrieval is evaluated in this section. The results are given in Table 7.15 for Raw String method and in Table 7.16 for Raw String with Answer Type method. In Table 7.15 and Table 7.16, *R.Docs* means the number of returned documents when query expansion is not applied, *R.Docs (QE)* means the number of returned documents when query expansion is applied, *Inc.Docs* means the increase in percent of the number of returned documents when query expansion is applied, *R.Sents* means the number of returned sentences when query expansion is not applied, *R.Sents (QE)* means the number of returned sentences when query expansion is applied, *Inc.Sents* means the increase in percent of the number of returned sentences when query expansion is applied.

Question Type	R.Docs	R.Docs (QE)	Inc.Docs	R.Sents	R.Sents (QE)	Inc.Sents
Author	2339	3897	66.61%	11292	14043	24.36%
Capital	2856	6605	131.27%	36180	42245	16.76%
DateOfBirth	2590	2598	0.31%	21268	21277	0.04%
DateOfDeath	2756	3035	10.12%	12574	12902	2.61%
Language	2945	5874	99.46%	34226	40271	17.66%
PlaceOfBirth	2509	4339	72.94%	20810	23065	10.84%
PlaceOfDeath	2764	2944	6.51%	12496	12737	1.93%
TOTAL	18759	29292	56.15%	148846	166540	11.89%

Table 7.15: Effect of query expansion on Document and Sentence Retrieval for Raw String method

According to Table 7.15 and Table 7.16, the results of Raw String method are very similar to the results of Raw String with Answer Type method. When query expansion is applied, both the number of returned documents and the number of returned sentences increase. *Author* question type has the maximum increase,

Question Type	R.Docs	R.Docs (QE)	Inc.Docs	R.Sents	R.Sents (QE)	Inc.Sents
Author	2339	3906	66.99%	11292	14058	24.50%
Capital	2856	6605	131.27%	36180	42245	16.76%
DateOfBirth	2590	2598	0.31%	21268	21277	0.04%
DateOfDeath	2756	3036	10.16%	12574	12903	2.62%
Language	2945	6006	103.94%	34226	40547	18.47%
PlaceOfBirth	2509	4339	72.94%	20810	23065	10.84%
PlaceOfDeath	2764	2944	6.51%	12496	12737	1.93%
TOTAL	18759	29434	56.91%	148846	166832	12.08%

Table 7.16: Effect of query expansion on Document and Sentence Retrieval for Raw String with Answer Type method

and *DateOfBirth* question type has the minimum increase. If a sentence contain the question phrase, it is returned by Sentence Retrieval phase. If the number of returned sentences increases for a question, the probability of answering the question also increases.

7.4.2 Effect of Query Expansion on the Returned Answer Sentences

The effect of query expansion on the number of returned sentences that contain answer phrase is evaluated in this section. These sentences are called *returned answer sentences*. The results are given in Table 7.17 for Raw String method and in Table 7.18 for Raw String with Answer Type method. Here *R.Sents* means the number of returned sentences when query expansion is not applied, *R.Sents (QE)* means the number of returned sentences when query expansion is applied, *A.Sents* means the number of returned answer sentences when query expansion is not applied, *A.Sents (QE)* means the number of returned answer sentences when query expansion is applied, *A/R* means the ratio between the returned answer sentences and returned sentences (A.Sents/R.Sents) when query expansion is not applied, and *A/R (QE)* means the ratio between the returned answer sentences and returned sentences (A.Sents (QE)/R.Sents (QE)) when query expansion is

applied.

Question Type	R.Sents	R.Sents (QE)	A.Sents	A.Sents (QE)	A/R	A/R (QE)
Author	11292	14043	666	921	0.059	0.066
Capital	36180	42245	1140	3852	0.032	0.091
DateOfBirth	21268	21277	157	157	0.007	0.007
DateOfDeath	12574	12902	116	124	0.009	0.010
Language	34226	40271	274	523	0.008	0.013
PlaceOfBirth	20810	23065	382	444	0.018	0.019
PlaceOfDeath	12496	12737	492	651	0.039	0.051
TOTAL	148846	166540	3227	6672	0.022	0.040

Table 7.17: Effect of query expansion on the returned answer sentences for Raw String method

Question Type	R.Sents	R.Sents (QE)	A.Sents	A.Sents (QE)	A/R	A/R (QE)
Author	11292	14058	666	921	0.059	0.066
Capital	36180	42245	1140	3852	0.032	0.091
DateOfBirth	21268	21277	157	157	0.007	0.007
DateOfDeath	12574	12903	116	124	0.009	0.010
Language	34226	40547	274	525	0.008	0.013
PlaceOfBirth	20810	23065	382	444	0.018	0.019
PlaceOfDeath	12496	12737	492	651	0.039	0.051
TOTAL	148846	166832	3227	6674	0.022	0.040

Table 7.18: Effect of query expansion on the returned answer sentences for Raw String with Answer Type method

According to Table 7.17 and Table 7.18, the ratio between the returned answer sentences and returned sentences increases to 0.040 from 0.022 when query expansion is applied. The A/R ratio of *DateOfBirth* question type does not change when query expansion is applied. There is a small increase in the A/R ratio of *DateOfDeath* and *PlaceOfBirth* question types when query expansion is applied. The A/R ratio of *Capital* question type has the maximum increase.

7.4.3 Effect of Query Expansion on Question Answering

The effect of query expansion on question answering is evaluated in this section. *MRR*, *Recall* and *Precision* values of the question answering system with and without query expansion are given in Table 7.19 for Raw String method and in Table 7.20 for Raw String with Answer Type method.

Question Type	MRR	MRR (QE)	Recall	Recall (QE)	Precision	Precision (QE)
Author	0.19	0.19	0.07	0.07	0.11	0.09
Capital	0.84	0.90	0.80	0.80	0.86	0.80
DateOfBirth	0.00	0.00	0.00	0.00	0.00	0.00
DateOfDeath	0.07	0.07	0.07	0.07	1.00	0.10
Language	0.70	0.64	0.60	0.60	0.64	0.60
PlaceOfBirth	0.07	0.05	0.07	0.00	0.20	0.00
PlaceOfDeath	0.07	0.07	0.07	0.07	1.00	0.50
TOTAL	0.28	0.27	0.24	0.23	0.57	0.40

Table 7.19: Results of query expansion for Raw String method

According to Table 7.19, *MRR*, *Recall* and *Precision* values of the system decrease when query expansion is applied for Raw String method. The probability of answer pattern matching increases when query expansion is applied. Answers are extracted by the matched answer patterns. Some of the extracted answers may be correct and some of them may be incorrect. The results show that the number of incorrect answers increases when query expansion is applied for Raw String method.

According to Table 7.20, *MRR* and *Recall* values of the system increase and *Precision* value decreases when query expansion is applied. Most of the incorrect answers are eliminated by checking the type of the answer. Query expansion increases the performance of the system for Raw String with Answer Type method.

Question Type	MRR	MRR (QE)	Recall	Recall (QE)	Precision	Precision (QE)
Author	0.40	0.47	0.40	0.47	1.00	0.78
Capital	0.87	1.00	0.87	1.00	1.00	1.00
DateOfBirth	0.00	0.00	0.00	0.00	0.00	0.00
DateOfDeath	0.07	0.07	0.07	0.07	1.00	0.20
Language	0.70	0.64	0.60	0.60	0.64	0.60
PlaceOfBirth	0.07	0.13	0.07	0.13	1.00	1.00
PlaceOfDeath	0.07	0.07	0.07	0.07	1.00	1.00
TOTAL	0.31	0.34	0.30	0.33	0.86	0.74

Table 7.20: Results of query expansion for Raw String with Answer Type method

7.5 Comparison

We compare the performance of our question answering system with the performances of the other question answering systems in this section. Table 7.21 shows the *MRR* values of different configurations of our QA system.

QA System	MRR
Raw	0.28
RawNE	0.31
Stemmed	0.29
StemmedNE	0.30
NeTagged	0.45
AllwithoutNE	0.56
AllwithNE	0.58
Raw (Re-ranked)	0.41
RawNE (Re-ranked)	0.39
Stemmed (Re-ranked)	0.33
StemmedNE (Re-ranked)	0.34
NeTagged (Re-ranked)	0.50
AllwithoutNE (Re-ranked)	0.61
AllwithNE (Re-ranked)	0.62
Raw (Query Expansion)	0.27
RawNE (Query Expansion)	0.34

Table 7.21: MRR results of our QA systems

TREC QA task is the major large scale evaluation environment for open-domain QA systems. It was first introduced in 1999 (TREC-8). 20 different organizations participated in the TREC-8 QA track. The TREC-8 document collection consist of mostly newspaper articles. 200 factoid questions were asked to participants. For each question, there was at least one document in the collection that explicitly answered the question. Participants returned a ranked list of five answer strings which were limited to 50 bytes. Human assessors read each answer string and assessed whether the answer string contains the correct answer. The MRR scores were computed for each participant. Minimum, maximum and average *MRR* values at TREC-8 QA Track [24] are given in Table 7.22.

The TREC-10 QA track included three separate tasks, the main task, the list task, and the context task. The main task was essentially the same as the task in the TREC-8 except for some questions there was no document in the collection that explicitly answered the question. We use the results of the main task to compare with our results. 500 factoid questions were asked to participants. The test questions of TREC-10 was much more difficult than the test questions of TREC-8. MRR scores were also computed for each participant. Minimum, maximum and average *MRR* values at TREC-10 QA Track [25] are given in Table 7.22.

Ephyra QA system [17] is one of the factoid QA systems which uses answer pattern matching approach. 700 questions were used to generate answer patterns. Ephyra QA system was evaluated on the 200 TREC-8 questions. A ranked list of up to five answers were returned and they were judged manually. The *MRR* score of Ephyra QA system is given in Table 7.22. Another QA system which uses answer pattern matching approach is presented in [16]. They used the questions of TREC-10 and AltaVista Web search engine to evaluate their system. Their question types were *Birthyear*, *Inventor*, *Discoverer*, *Location*, etc. The *MRR* score of their system is also given in Table 7.22.

BayBilmis QA System [1] was developed for Turkish QA. They used 524 test questions from TREC-9 and TREC-10 to evaluate BayBilmis QA system. The questions were translated from English to Turkish. The *MRR* value of BayBilmis

is given in Table 7.22.

QA System	MRR
TREC-8 (max)	0.66
TREC-8 (avg)	0.25
TREC-8 (min)	0.02
TREC-10 (max)	0.68
TREC-10 (avg)	0.39
TREC-10 (min)	0.27
Ephyra	0.40
QA System [16]	0.57
BayBilmiş	0.31

Table 7.22: MRR results of QA systems

According to Table 7.21 and Table 7.22, all of the configurations of our system have higher *MRR* value than the minimum *MRR* values of TREC-8 and TREC-10. *NeTagged*, *AllwithoutNE*, and *AllwithNE* configurations have higher *MRR* values than the average *MRR* values of TREC-8 and TREC-10. Especially, the *MRR* values of *AllwithoutNE (Re-ranked)* and *AllwithNE (Re-ranked)* configurations are close to the maximum *MRR* values of TREC-8 and TREC-10. *AllwithoutNE*, *AllwithNE*, *AllwithoutNE (Re-ranked)* and *AllwithNE (Re-ranked)* configurations of our system have better *MRR* values than Ephyra and BayBilmis. The *MRR* values of these configurations and the *MRR* value of the QA system presented in [16] are similar.

Chapter 8

Conclusion and Future Work

In this thesis, we have investigated the potential of answer pattern matching technique for Turkish Factoid Question Answering (QA). Answer pattern matching technique uses textual patterns to extract answers. Since the technique has been used successfully for English Factoid QA such as [16], [17], [22], we therefore decided to implement it by applying various answer pattern extraction methods for Turkish Factoid QA. These methods are Raw String, Raw String with Answer Type, Stemmed String, Stemmed String with Answer Type, and Named Entity Tagged String methods. These methods are compared according to *MRR*, *Recall* and *Precision* scores:

- The scores of Stemmed String methods are slightly better than the scores of Raw String methods so stemming slightly improves the performance of the system.
- The scores of Raw and Stemmed String with Answer Type methods are better than the scores of Raw and Stemmed String methods so checking the answer type improves the performance of the system.
- Named Entity Tagged String method has the best scores so replacing words with their named entity tags improves the performance of the system.

We have implemented an answer re-ranking approach based on Frequency Counting technique. The technique relies on correct answers to appear more frequently than incorrect answers. The technique increases the *MRR* and *Recall* scores. Answer re-ranking approach has a good impact on the performance of the system.

We have implemented a query expansion approach using answer patterns. We use the most reliable answer patterns to extend the queries. The number of sentences containing answer phrase increases when query expansion is applied. The scores of Raw String with Answer Type method increase when query expansion is applied. However, the scores of Raw String method decrease when query expansion is applied.

An important limitation of our answer pattern matching approach is that an answer pattern can include only one question phrase. It does not work for the question types which have multiple question phrases, possibly apart from each other. For example, in order to answer the questions which ask the president of a country at an exact date ("2003 yılında Türkiye'nin cumhurbaşkanı kimdi?"), the answer patterns of that question type should include two question phrases, one question phrase for the name of the country ("Türkiye") and one question phrase for the date ("2003"). The answer pattern "<Q1>'nin <Q2> yılındaki cumhurbaşkanı <A>" can be used to answer this type of questions.

Another drawback is that the answer patterns cannot handle long-distance relationships between the question phrase and the answer phrase. For example, the answer pattern "<Q>'nin başkenti <A>" cannot find the answer in the sentence "Türkiye'nin başkenti ve ikinci büyük şehri olan Ankara, İç Anadolu Bölgesindedir." However, since the factoid information is usually replicated and expressed in many different forms across the Web, it is feasible to find an instance of the answer patterns with high probability.

Another issue is that answer phrases could be written in many ways. For instance, a date can be written in different forms. (Mustafa Kemal Atatürk's birth date can be written as "1881", "19 Mayıs 1881", "19.05.1881", etc.) The same issue also applies to question phrases. For example, the names of persons

can be written in different forms. (Mustafa Kemal Atatürk could be written as "Atatürk", "Mustafa Kemal", "M. K. Atatürk", etc.)

Investigating the potential of more generic answer patterns is left as a future work. Stemmed String and Named Entity Tagged String methods extract more generic answer patterns from Raw String methods and they achieved better results. More generic answer patterns can be extracted by using linguistic techniques such as phrase chunking and morphological analysis. We believe that combining different answer processing techniques can improve the *Recall* significantly. When there is no answer pattern match, the system returns no answer. Therefore, the combination of backup techniques is reasonable.

Bibliography

[1] M. F. Amasyalı and B. Diri. Bir soru cevaplama sistemi: Baybilmiş. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 2005.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, 1999.

[3] Bing. Bing web search engine. http://www.bing.com/.

[4] E. Brill, S. Dumais, and M. Banko. An analysis of the askMSR question-answering system. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, 2002.

[5] CLEF. Cross Language Evaluation Forum. http://www.clef-campaign.org/.

[6] O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134, 2005.

[7] C. Fellbaum. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

[8] B. Green, C. Chomsky, and K. Laughery. BASEBALL: An automatic question answerer. *Proceedings of the Western Joint Computer Conference*, 1961.

[9] K. Hammond. FAQ finder: a case-based approach to knowledge navigation. *Proceedings of the 11th conference on artificial intelligence for applications*, 1995.

[10] E. Hovy, U. Hermjakob, and D. Ravichandran. A question/answer typology with surface text patterns. *In Proceedings of the Second International Conference on Human Language Technology Research*, 2002.

[11] V. Jijkoun, G. Mishne, and M. de Rijke. Preprocessing documents to answer dutch questions. *Belgian-Netherlands Conference on Artificial Intelligence*, 2003.

[12] D. Jurafsky and J. H. Martin. *Speech and Language Processing*. Prentice Hall, 2 edition, 2008.

[13] B. Katz, g. Borchardt, and S. Felshin. Syntactic and semantic decomposition strategies for question answering from multiple resources. *In Proceedings of the AAAI 2005 Workshop on Inference for Textual Question Answering*, 2005.

[14] D. Modolvan, M. Pasca, S. Harabagiu, and M. Surdeanu. Performance issues and error analysis in an open-domain question answering system. *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

[15] NTCIR. NII Test Collection for IR Systems. http://research.nii.ac.jp/ntcir/.

[16] D. Ravichandran and E. Hovy. Learning surface text patterns for a question answering system. *In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2001.

[17] N. Schlaefer and P. Gieselmann. A pattern learning approach to question answering within the ephyra framework. *In Proceedings of the 9th International Conference on Text, Speech and Dialogue*, 2006.

[18] N. Schlaefer, G. Sautter, T. Schaaf, and A. Waibel. The ephyra qa system at trec 2006. *In Proceedings of the 15th Text Retrieval Conference*, 2006.

[19] P. Schone, G. Ciany, R. Cutts, J. Mayfield, and T. Smith. QACTIS-based question answering at TREC-2005. *In Proceedings of the 14th Text Retrieval Conference*, 2005.

[20] R. Simmons. Answering english questions by computer: A survey. *Communications of the ACM*, 1965.

[21] E. Sneiders. Automated FAQ answering: continued experience with shallow language understanding. *Proceedings of the 1999 AAAI Fall Symposium on Question Answering Systems*, 1999.

[22] M. Soubbotin and S. Soubbotin. Patterns of potential answer expressions as clues to the right answer. *In Proceedings of the 10th Text Retrieval Conference*, 2001.

[23] TREC. Text REtrieval Conference. http://trec.nist.gov/.

[24] E. M. Voorhees. The TREC-8 question answering track report. *In Proceedings of the 8th Text Retrieval Conference*, 1999.

[25] E. M. Voorhees. Overview of the TREC 2001 question answering track. *In Proceedings of the 10th Text Retrieval Conference*, 2001.

[26] E. M. Voorhees. Overview of the TREC 2003 question answering track. *In Proceedings of the 12th Text Retrieval Conference*, 2003.

[27] WolframAlpha. Computational Knowledge Engine. http://www.wolframalpha.com/.

[28] W. Woods. Progress in natural language understanding: An application to lunar geology. *Proceedings of the AFIPS Conference*, 1973.

[29] M. Wu, X. Zheng, M. Duan, T.Liu, and T. Strzalkowski. Question answering by pattern matching, web-proofing, semantic form proofing. *In Proceedings of the 12th Text Retrieval Conference*, 2003.

Appendix A

Question-Answer Pairs

Question-Answer pairs used for seven question types are given in this section. These question types are *Author*, *Capital*, *DateOfBirth*, *DateOfDeath*, *Language*, *PlaceOfBirth*, and *PlaceOfDeath*. Answer patterns are learned for each question type by using the first 15 question-answer pairs. The system is evaluated by using the remaining 15 question-answer pairs.

#	Question	Answer
1	cevdet bey ve oğulları	orhan pamuk
2	benim adım kırmızı	orhan pamuk
3	çalıkuşu	reşat nuri güntekin
4	dudaktan kalbe	reşat nuri güntekin
5	gün olur asra bedel	cengiz aytmatov
6	ateşten gömlek	halide edip adıvar
7	sinekli bakkal	halide edip adıvar
8	vurun kahpeye	halide edip adıvar
9	memleketimden insan manzaraları	nazım hikmet
10	tahir ile zühre	nazım hikmet
11	kırmızı bisiklet	can dündar
12	yağmurdan sonra	can dündar
13	otuz beş yaş	cahit sıtkı tarancı
14	puslu kıtalar atlası	ihsan oktay anar
15	şu çılgın türkler	turgut özakman
16	romantika	turgut özakman
17	ince memed	yaşar kemal
18	ölmez otu	yaşar kemal
19	yılanı öldürseler	yaşar kemal
20	karıncanın su içtiği	yaşar kemal
21	tutunamayanlar	oğuz atay
22	tehlikeli oyunlar	oğuz atay
23	korkuyu beklerken	oğuz atay
24	huzur	ahmet hamdi tanpınar
25	saatleri ayarlama enstitüsü	ahmet hamdi tanpınar
26	yaban	yakup kadri karaosmanoğlu
27	fikrimin ince gülü	adalet ağaoğlu
28	kuyucaklı yusuf	sabahattin ali
29	mai ve siyah	halit ziya uşaklıgil
30	mor	inci aral

Table A.1: Question-Answer pairs for *Author* question type

#	Question	Answer
1	gürcistan	tiflis
2	türkiye	ankara
3	almanya	berlin
4	norveç	oslo
5	fransa	paris
6	yunanistan	atina
7	suriye	şam
8	çin	pekin
9	mısır	kahire
10	romanya	bükreş
11	rusya	moskova
12	italya	roma
13	ispanya	madrid
14	finlandiya	helsinki
15	japonya	tokyo
16	hollanda	amsterdam
17	azerbaycan	bakü
18	macaristan	budapeşte
19	ermenistan	erivan
20	bulgaristan	sofya
21	ingiltere	londra
22	avusturya	viyana
23	belçika	brüksel
24	iran	tahran
25	sırbistan	belgrad
26	kırgızistan	bişkek
27	danimarka	kopenhag
28	polonya	varşova
29	hırvatistan	zagrep
30	özbekistan	taşkent

Table A.2: Question-Answer pairs for *Capital* question type

#	Question	Answer
1	mustafa kemal atatürk	19 mayıs 1881
2	ismet inönü	24 eylül 1884
3	ismail cem ipekçi	15 şubat 1940
4	turgut özal	13 ekim 1927
5	uğur mumcu	22 ağustos 1942
6	sadri alışık	5 nisan 1925
7	kemal sunal	11 kasım 1944
8	ali babacan	4 nisan 1967
9	necip fazıl kısakürek	26 mayıs 1904
10	mehmet akif ersoy	20 aralık 1873
11	recep tayyip erdoğan	26 şubat 1954
12	oktay sinanoğlu	25 şubat 1935
13	fazıl hüsnü dağlarca	26 ağustos 1914
14	deniz baykal	20 temmuz 1938
15	filiz akın	2 ocak 1943
16	şener şen	26 aralık 1941
17	yılmaz güney	1 nisan 1937
18	barış manço	2 ocak 1943
19	barış akarsu	29 haziran 1979
20	bülent ecevit	28 mayıs 1925
21	fazıl say	14 ocak 1970
22	ara güler	16 ağustos 1928
23	idil biret	21 kasım 1941
24	arif mardin	15 mart 1932
25	ahmet ertegün	31 temmuz 1923
26	mehmet okur	26 mayıs 1979
27	cahit arf	1910
28	orhan pamuk	7 haziran 1952
29	abdullah gül	29 ekim 1950
30	reşat nuri güntekin	25 kasım 1889

Table A.3: Question-Answer pairs for *DateOfBirth* question type

#	Question	Answer
1	mustafa kemal atatürk	10 kasım 1938
2	ismet inönü	25 aralık 1973
3	ismail cem ipekçi	24 ocak 2007
4	turgut özal	17 nisan 1993
5	uğur mumcu	24 ocak 1993
6	sadri alışık	18 mart 1995
7	kemal sunal	3 temmuz 2000
8	necip fazıl kısakürek	25 mayıs 1983
9	mehmet akif ersoy	27 aralık 1936
10	hüseyin rahmi gürpınar	8 mart 1944
11	yılmaz güney	9 eylül 1984
12	barış manço	1 şubat 1999
13	barış akarsu	4 temmuz 2007
14	bülent ecevit	5 kasım 2006
15	deniz gezmiş	6 mayıs 1972
16	ayhan şahenk	1 nisan 2001
17	gaffar okan	24 ocak 2001
18	necip hablemitoğlu	18 aralık 2002
19	sakıp sabancı	10 nisan 2004
20	erdal inönü	31 ekim 2007
21	arif mardin	25 haziran 2006
22	ahmet ertegün	14 aralık 2006
23	cahit arf	26 aralık 1997
24	reşat nuri güntekin	7 aralık 1956
25	cahit sıtkı tarancı	13 ekim 1956
26	sabiha gökçen	22 mart 2001
27	osman yağmurdereli	1 ağustos 2008
28	üzeyir garih	25 ağustos 2001
29	abidin dino	7 aralık 1993
30	alparslan türkeş	4 nisan 1997

Table A.4: Question-Answer pairs for *DateOfDeath* question type

#	Question	Answer
1	türkiye	türkçe
2	fransa	fransızca
3	yunanistan	yunanca
4	romanya	romence
5	rusya	rusça
6	suriye	arapça
7	ispanya	ispanyolca
8	finlandiya	fince
9	japonya	japonca
10	özbekistan	özbekçe
11	mısır	arapça
12	macaristan	macarca
13	ermenistan	ermenice
14	gürcistan	gürcüce
15	almanya	almanca
16	avusturya	almanca
17	azerbaycan	azerice
18	bulgaristan	bulgarca
19	ingiltere	ingilizce
20	iran	farsça
21	hırvatistan	hırvatça
22	küba	ispanyolca
23	tunus	arapça
24	moğolistan	moğolca
25	hindistan	hintçe
26	arjantin	ispanyolca
27	kolombiya	ispanyolca
28	peru	ispanyolca
29	yemen	arapça
30	kuveyt	arapça

Table A.5: Question-Answer pairs for *Language* question type

#	Question	Answer
1	mustafa kemal atatürk	selanik
2	ismet inönü	izmir
3	ismail cem ipekçi	istanbul
4	turgut özal	malatya
5	uğur mumcu	kırşehir
6	sadri alışık	istanbul
7	kemal sunal	malatya
8	ali babacan	ankara
9	necip fazıl kısakürek	istanbul
10	mehmet akif ersoy	istanbul
11	recep tayyip erdoğan	istanbul
12	oktay sinanoğlu	bari
13	fazıl hüsnü dağlarca	istanbul
14	deniz baykal	antalya
15	filiz akın	ankara
16	şener şen	adana
17	yılmaz güney	adana
18	barış manço	istanbul
19	barış akarsu	zonguldak
20	bülent ecevit	istanbul
21	fazıl say	ankara
22	idil biret	ankara
23	arif mardin	istanbul
24	ahmet ertegün	istanbul
25	mehmet okur	yalova
26	cahit arf	selanik
27	orhan pamuk	istanbul
28	abdullah gül	kayseri
29	reşat nuri güntekin	istanbul
30	cahit sıtkı tarancı	diyarbakır

Table A.6: Question-Answer pairs for *PlaceOfBirth* question type

#	Question	Answer
1	mustafa kemal atatürk	istanbul
2	ismet inönü	ankara
3	ismail cem ipekçi	istanbul
4	turgut özal	ankara
5	uğur mumcu	ankara
6	sadri alışık	istanbul
7	kemal sunal	istanbul
8	necip fazıl kısakürek	istanbul
9	mehmet akif ersoy	istanbul
10	hüseyin rahmi gürpınar	istanbul
11	yılmaz güney	paris
12	barış manço	istanbul
13	barış akarsu	bodrum
14	bülent ecevit	ankara
15	deniz gezmiş	ankara
16	ayhan şahenk	istanbul
17	gaffar okan	diyarbakır
18	necip hablemitoğlu	ankara
19	sakıp sabancı	istanbul
20	erdal inönü	houston
21	arif mardin	new york
22	ahmet ertegün	new york
23	cahit arf	istanbul
24	reşat nuri güntekin	londra
25	cahit sıtkı tarancı	viyana
26	sabiha gökçen	ankara
27	osman yağmurdereli	istanbul
28	üzeyir garih	istanbul
29	abidin dino	paris
30	alparslan türkeş	ankara

Table A.7: Question-Answer pairs for *PlaceOfDeath* question type

Appendix B

Evaluation Results

Results of answer pattern extraction methods at different thresholds are given in Table B.1. *#Q* means the number of test questions, *#A* means the number of returned answers, *#CA* means the number of correct answers in the first position, *Prec.* means precision, *Rec.* means recall, and *#AP* means the number of answer patterns. The first column shows the question types and the last row shows the total results. We use the same table template in the presentation of the results of the other answer pattern extraction methods.

Question Type	#Q	#A	#CA	MRR	Prec.	Rec.	#AP
Threshold: 0.55							
Raw							
Author	15	12	1	0.21	0.08	0.07	360
Capital	15	15	13	0.91	0.87	0.87	3288
DateOfBirth	15	7	3	0.20	0.43	0.20	90
DateOfDeath	15	5	2	0.13	0.40	0.13	390
Language	15	14	9	0.70	0.64	0.60	2164
PlaceOfBirth	15	9	2	0.17	0.22	0.13	716
PlaceOfDeath	15	4	1	0.07	0.25	0.07	861
TOTAL	105	66	31	0.34	0.47	0.30	7869
RawNE							

Author	15	7	6	0.42	0.86	0.40	360
Capital	15	13	13	0.87	1.00	0.87	2654
DateOfBirth	15	3	3	0.20	1.00	0.20	90
DateOfDeath	15	2	2	0.13	1.00	0.13	390
Language	15	14	9	0.70	0.64	0.60	2164
PlaceOfBirth	15	1	1	0.07	1.00	0.07	516
PlaceOfDeath	15	1	1	0.07	1.00	0.07	617
TOTAL	105	41	35	0.35	0.85	0.33	6791
Stemmed							
Author	15	13	1	0.16	0.08	0.07	196
Capital	15	15	14	0.94	0.93	0.93	2695
DateOfBirth	15	7	3	0.20	0.43	0.20	76
DateOfDeath	15	6	1	0.07	0.17	0.07	316
Language	15	14	9	0.69	0.64	0.60	2043
PlaceOfBirth	15	11	1	0.07	0.09	0.07	514
PlaceOfDeath	15	4	1	0.10	0.25	0.07	721
TOTAL	105	70	30	0.32	0.43	0.29	6561
StemmedNE							
Author	15	5	5	0.33	1.00	0.33	196
Capital	15	13	13	0.87	1.00	0.87	2310
DateOfBirth	15	3	3	0.20	1.00	0.20	76
DateOfDeath	15	1	1	0.07	1.00	0.07	316
Language	15	14	9	0.69	0.64	0.60	2043
PlaceOfBirth	15	1	1	0.07	1.00	0.07	395
PlaceOfDeath	15	2	2	0.13	1.00	0.13	538
TOTAL	105	39	34	0.34	0.87	0.32	5874
NETagged							
Author	15	14	11	0.74	0.79	0.73	240
Capital	15	13	13	0.87	1.00	0.87	2375
DateOfBirth	15	11	8	0.56	0.73	0.53	74
DateOfDeath	15	5	4	0.29	0.80	0.27	235

Language	15	0	0	0.00	0.00	0.00	0
PlaceOfBirth	15	13	13	0.87	1.00	0.87	410
PlaceOfDeath	15	6	2	0.13	0.33	0.13	522
TOTAL	105	62	51	0.49	0.82	0.49	3856
AllMethodsWithoutNE							
Author	15	15	10	0.71	0.67	0.67	796
Capital	15	15	14	0.94	0.93	0.93	8358
DateOfBirth	15	13	8	0.56	0.62	0.53	240
DateOfDeath	15	12	4	0.30	0.33	0.27	941
Language	15	15	9	0.66	0.60	0.60	4207
PlaceOfBirth	15	15	10	0.74	0.67	0.67	1640
PlaceOfDeath	15	9	2	0.17	0.22	0.13	2104
TOTAL	105	94	57	0.58	0.61	0.54	18286
AllMethodsWithNE							
Author	15	14	11	0.74	0.79	0.73	796
Capital	15	14	14	0.93	1.00	0.93	7339
DateOfBirth	15	11	8	0.56	0.73	0.53	240
DateOfDeath	15	8	5	0.35	0.63	0.33	941
Language	15	15	9	0.66	0.60	0.60	4207
PlaceOfBirth	15	14	13	0.87	0.93	0.87	1321
PlaceOfDeath	15	7	3	0.20	0.43	0.20	1677
TOTAL	105	83	63	0.62	0.76	0.60	16521
Threshold: 0.6							
Raw							
Author	15	12	1	0.19	0.08	0.07	358
Capital	15	15	13	0.91	0.87	0.87	3282
DateOfBirth	15	1	0	0.00	0.00	0.00	88
DateOfDeath	15	5	2	0.13	0.40	0.13	390
Language	15	14	9	0.70	0.64	0.60	2161
PlaceOfBirth	15	9	2	0.17	0.22	0.13	714
PlaceOfDeath	15	4	1	0.07	0.25	0.07	860

TOTAL	105	60	28	0.31	0.47	0.27	7853
RawNE							
Author	15	7	6	0.40	0.86	0.40	358
Capital	15	13	13	0.87	1.00	0.87	2649
DateOfBirth	15	0	0	0.00	0.00	0.00	88
DateOfDeath	15	2	2	0.13	1.00	0.13	390
Language	15	14	9	0.70	0.64	0.60	2161
PlaceOfBirth	15	1	1	0.07	1.00	0.07	514
PlaceOfDeath	15	1	1	0.07	1.00	0.07	616
TOTAL	105	38	32	0.32	0.84	0.30	6776
Stemmed							
Author	15	13	1	0.16	0.08	0.07	196
Capital	15	15	14	0.94	0.93	0.93	2690
DateOfBirth	15	1	0	0.00	0.00	0.00	74
DateOfDeath	15	6	1	0.07	0.17	0.07	315
Language	15	14	9	0.69	0.64	0.60	2041
PlaceOfBirth	15	11	1	0.07	0.09	0.07	512
PlaceOfDeath	15	4	1	0.10	0.25	0.07	720
TOTAL	105	64	27	0.29	0.42	0.26	6548
StemmedNE							
Author	15	5	5	0.33	1.00	0.33	196
Capital	15	13	13	0.87	1.00	0.87	2306
DateOfBirth	15	0	0	0.00	0.00	0.00	74
DateOfDeath	15	1	1	0.07	1.00	0.07	315
Language	15	14	9	0.69	0.64	0.60	2041
PlaceOfBirth	15	1	1	0.07	1.00	0.07	392
PlaceOfDeath	15	2	2	0.13	1.00	0.13	538
TOTAL	105	36	31	0.31	0.86	0.30	5862
NETagged							
Author	15	14	11	0.74	0.79	0.73	236
Capital	15	13	13	0.87	1.00	0.87	2363

DateOfBirth	15	11	8	0.56	0.73	0.53	74
DateOfDeath	15	5	4	0.29	0.80	0.27	233
Language	15	0	0	0.00	0.00	0.00	0
PlaceOfBirth	15	13	13	0.87	1.00	0.87	408
PlaceOfDeath	15	6	2	0.13	0.33	0.13	522
TOTAL	105	62	51	0.49	0.82	0.49	3836
AllMethodsWithoutNE							
Author	15	15	10	0.71	0.67	0.67	790
Capital	15	15	14	0.94	0.93	0.93	8335
DateOfBirth	15	11	8	0.56	0.73	0.53	236
DateOfDeath	15	12	4	0.30	0.33	0.27	938
Language	15	15	9	0.66	0.60	0.60	4202
PlaceOfBirth	15	15	10	0.74	0.67	0.67	1634
PlaceOfDeath	15	9	2	0.17	0.22	0.13	2102
TOTAL	105	92	57	0.58	0.62	0.54	18237
AllMethodsWithNE							
Author	15	14	11	0.74	0.79	0.73	790
Capital	15	14	14	0.93	1.00	0.93	7318
DateOfBirth	15	11	8	0.56	0.73	0.53	236
DateOfDeath	15	8	5	0.35	0.63	0.33	938
Language	15	15	9	0.66	0.60	0.60	4202
PlaceOfBirth	15	14	13	0.87	0.93	0.87	1314
PlaceOfDeath	15	7	3	0.20	0.43	0.20	1676
TOTAL	105	83	63	0.62	0.76	0.60	16474
Threshold: 0.65							
Raw							
Author	15	11	1	0.19	0.09	0.07	344
Capital	15	15	13	0.91	0.87	0.87	3245
DateOfBirth	15	1	0	0.00	0.00	0.00	86
DateOfDeath	15	4	2	0.13	0.50	0.13	382
Language	15	14	9	0.70	0.64	0.60	2143

PlaceOfBirth	15	7	1	0.10	0.14	0.07	708
PlaceOfDeath	15	4	1	0.07	0.25	0.07	853
TOTAL	105	56	27	0.30	0.48	0.26	7761
RawNE							
Author	15	7	6	0.40	0.86	0.40	344
Capital	15	13	13	0.87	1.00	0.87	2614
DateOfBirth	15	0	0	0.00	0.00	0.00	86
DateOfDeath	15	2	2	0.13	1.00	0.13	382
Language	15	14	9	0.70	0.64	0.60	2143
PlaceOfBirth	15	1	1	0.07	1.00	0.07	509
PlaceOfDeath	15	1	1	0.07	1.00	0.07	609
TOTAL	105	38	32	0.32	0.84	0.30	6687
Stemmed							
Author	15	12	1	0.16	0.08	0.07	188
Capital	15	15	14	0.94	0.93	0.93	2668
DateOfBirth	15	1	0	0.00	0.00	0.00	73
DateOfDeath	15	3	1	0.07	0.33	0.07	309
Language	15	13	9	0.69	0.69	0.60	2020
PlaceOfBirth	15	11	1	0.07	0.09	0.07	507
PlaceOfDeath	15	4	1	0.10	0.25	0.07	716
TOTAL	105	59	27	0.29	0.46	0.26	6481
StemmedNE							
Author	15	5	5	0.33	1.00	0.33	188
Capital	15	13	13	0.87	1.00	0.87	2281
DateOfBirth	15	0	0	0.00	0.00	0.00	73
DateOfDeath	15	1	1	0.07	1.00	0.07	309
Language	15	13	9	0.69	0.69	0.60	2020
PlaceOfBirth	15	1	1	0.07	1.00	0.07	387
PlaceOfDeath	15	2	2	0.13	1.00	0.13	533
TOTAL	105	35	31	0.31	0.89	0.30	5791
NETagged							

Author	15	13	10	0.68	0.77	0.67	226
Capital	15	13	13	0.87	1.00	0.87	2347
DateOfBirth	15	9	8	0.56	0.89	0.53	73
DateOfDeath	15	4	4	0.27	1.00	0.27	232
Language	15	0	0	0.00	0.00	0.00	0
PlaceOfBirth	15	12	12	0.80	1.00	0.80	405
PlaceOfDeath	15	3	2	0.13	0.67	0.13	518
TOTAL	105	54	49	0.47	0.91	0.47	3801
AllMethodsWithoutNE							
Author	15	15	10	0.71	0.67	0.67	758
Capital	15	15	14	0.94	0.93	0.93	8260
DateOfBirth	15	10	8	0.56	0.80	0.53	232
DateOfDeath	15	11	4	0.29	0.36	0.27	923
Language	15	14	9	0.66	0.64	0.60	4163
PlaceOfBirth	15	15	10	0.70	0.67	0.67	1620
PlaceOfDeath	15	7	2	0.17	0.29	0.13	2087
TOTAL	105	87	57	0.57	0.66	0.54	18043
AllMethodsWithNE							
Author	15	14	11	0.74	0.79	0.73	758
Capital	15	14	14	0.93	1.00	0.93	7242
DateOfBirth	15	9	8	0.56	0.89	0.53	232
DateOfDeath	15	8	5	0.33	0.63	0.33	923
Language	15	14	9	0.66	0.64	0.60	4163
PlaceOfBirth	15	13	12	0.80	0.92	0.80	1301
PlaceOfDeath	15	4	3	0.20	0.75	0.20	1660
TOTAL	105	76	62	0.60	0.82	0.59	16279
Threshold: 0.7							
Raw							
Author	15	10	1	0.19	0.10	0.07	247
Capital	15	15	13	0.91	0.87	0.87	1189
DateOfBirth	15	1	0	0.00	0.00	0.00	31

DateOfDeath	15	1	1	0.07	1.00	0.07	185
Language	15	14	9	0.70	0.64	0.60	599
PlaceOfBirth	15	5	1	0.07	0.20	0.07	371
PlaceOfDeath	15	1	1	0.07	1.00	0.07	470
TOTAL	105	47	26	0.29	0.55	0.25	3092
RawNE							
Author	15	7	6	0.40	0.86	0.40	247
Capital	15	13	13	0.87	1.00	0.87	925
DateOfBirth	15	0	0	0.00	0.00	0.00	31
DateOfDeath	15	1	1	0.07	1.00	0.07	185
Language	15	14	9	0.70	0.64	0.60	599
PlaceOfBirth	15	1	1	0.07	1.00	0.07	275
PlaceOfDeath	15	1	1	0.07	1.00	0.07	332
TOTAL	105	37	31	0.31	0.84	0.30	2594
Stemmed							
Author	15	11	1	0.16	0.09	0.07	143
Capital	15	15	14	0.93	0.93	0.93	958
DateOfBirth	15	1	0	0.00	0.00	0.00	27
DateOfDeath	15	3	1	0.07	0.33	0.07	156
Language	15	13	9	0.69	0.69	0.60	554
PlaceOfBirth	15	5	1	0.07	0.20	0.07	273
PlaceOfDeath	15	2	1	0.10	0.50	0.07	391
TOTAL	105	50	27	0.29	0.54	0.26	2502
StemmedNE							
Author	15	5	5	0.33	1.00	0.33	143
Capital	15	12	12	0.80	1.00	0.80	783
DateOfBirth	15	0	0	0.00	0.00	0.00	27
DateOfDeath	15	1	1	0.07	1.00	0.07	156
Language	15	13	9	0.69	0.69	0.60	554
PlaceOfBirth	15	1	1	0.07	1.00	0.07	211
PlaceOfDeath	15	2	2	0.13	1.00	0.13	288

TOTAL	105	34	30	0.30	0.88	0.29	2162
NETagged							
Author	15	13	10	0.68	0.77	0.67	176
Capital	15	13	13	0.87	1.00	0.87	832
DateOfBirth	15	9	8	0.56	0.89	0.53	33
DateOfDeath	15	4	4	0.27	1.00	0.27	126
Language	15	0	0	0.00	0.00	0.00	0
PlaceOfBirth	15	12	12	0.80	1.00	0.80	206
PlaceOfDeath	15	3	2	0.13	0.67	0.13	281
TOTAL	105	54	49	0.47	0.91	0.47	1654
AllMethodsWithoutNE							
Author	15	15	10	0.71	0.67	0.67	566
Capital	15	15	14	0.94	0.93	0.93	2979
DateOfBirth	15	10	8	0.56	0.80	0.53	91
DateOfDeath	15	9	3	0.22	0.33	0.20	467
Language	15	14	9	0.66	0.64	0.60	1153
PlaceOfBirth	15	15	10	0.70	0.67	0.67	850
PlaceOfDeath	15	4	2	0.17	0.50	0.13	1142
TOTAL	105	82	56	0.57	0.68	0.53	7248
AllMethodsWithNE							
Author	15	14	11	0.74	0.79	0.73	566
Capital	15	14	14	0.93	1.00	0.93	2540
DateOfBirth	15	9	8	0.56	0.89	0.53	91
DateOfDeath	15	6	4	0.27	0.67	0.27	467
Language	15	14	9	0.66	0.64	0.60	1153
PlaceOfBirth	15	12	12	0.80	1.00	0.80	692
PlaceOfDeath	15	4	3	0.20	0.75	0.20	901
TOTAL	105	73	61	0.59	0.84	0.58	6410
Threshold: 0.75							
Raw							
Author	15	9	1	0.19	0.11	0.07	241

Capital	15	14	12	0.84	0.86	0.80	1181
DateOfBirth	15	0	0	0.00	0.00	0.00	30
DateOfDeath	15	1	1	0.07	1.00	0.07	183
Language	15	14	9	0.70	0.64	0.60	597
PlaceOfBirth	15	5	1	0.07	0.20	0.07	370
PlaceOfDeath	15	1	1	0.07	1.00	0.07	467
TOTAL	105	44	25	0.28	0.57	0.24	3069
RawNE							
Author	15	6	6	0.40	1.00	0.40	241
Capital	15	13	13	0.87	1.00	0.87	919
DateOfBirth	15	0	0	0.00	0.00	0.00	30
DateOfDeath	15	1	1	0.07	1.00	0.07	183
Language	15	14	9	0.70	0.64	0.60	597
PlaceOfBirth	15	1	1	0.07	1.00	0.07	273
PlaceOfDeath	15	1	1	0.07	1.00	0.07	327
TOTAL	105	36	31	0.31	0.86	0.30	2570
Stemmed							
Author	15	10	1	0.16	0.10	0.07	140
Capital	15	15	14	0.93	0.93	0.93	947
DateOfBirth	15	0	0	0.00	0.00	0.00	26
DateOfDeath	15	3	1	0.07	0.33	0.07	154
Language	15	13	9	0.69	0.69	0.60	551
PlaceOfBirth	15	4	1	0.07	0.25	0.07	272
PlaceOfDeath	15	2	1	0.10	0.50	0.07	388
TOTAL	105	47	27	0.29	0.57	0.26	2478
StemmedNE							
Author	15	5	5	0.33	1.00	0.33	140
Capital	15	12	12	0.80	1.00	0.80	776
DateOfBirth	15	0	0	0.00	0.00	0.00	26
DateOfDeath	15	1	1	0.07	1.00	0.07	154
Language	15	13	9	0.69	0.69	0.60	551

PlaceOfBirth	15	1	1	0.07	1.00	0.07	209
PlaceOfDeath	15	2	2	0.13	1.00	0.13	285
TOTAL	105	34	30	0.30	0.88	0.29	2141
NETagged							
Author	15	11	10	0.68	0.91	0.67	170
Capital	15	13	13	0.87	1.00	0.87	830
DateOfBirth	15	9	8	0.56	0.89	0.53	33
DateOfDeath	15	4	4	0.27	1.00	0.27	125
Language	15	0	0	0.00	0.00	0.00	0
PlaceOfBirth	15	10	10	0.67	1.00	0.67	205
PlaceOfDeath	15	3	2	0.13	0.67	0.13	278
TOTAL	105	50	47	0.45	0.94	0.45	1641
AllMethodsWithoutNE							
Author	15	15	10	0.71	0.67	0.67	551
Capital	15	15	14	0.94	0.93	0.93	2958
DateOfBirth	15	9	8	0.56	0.89	0.53	89
DateOfDeath	15	9	3	0.22	0.33	0.20	462
Language	15	14	9	0.66	0.64	0.60	1148
PlaceOfBirth	15	15	10	0.68	0.67	0.67	847
PlaceOfDeath	15	4	2	0.17	0.50	0.13	1133
TOTAL	105	81	56	0.56	0.69	0.53	7188
AllMethodsWithNE							
Author	15	11	10	0.68	0.91	0.67	551
Capital	15	14	14	0.93	1.00	0.93	2525
DateOfBirth	15	9	8	0.56	0.89	0.53	89
DateOfDeath	15	6	4	0.27	0.67	0.27	462
Language	15	14	9	0.66	0.64	0.60	1148
PlaceOfBirth	15	11	11	0.73	1.00	0.73	687
PlaceOfDeath	15	4	3	0.20	0.75	0.20	890
TOTAL	105	69	59	0.58	0.86	0.56	6352
Threshold: 0.8							

Raw							
Author	15	4	1	0.07	0.25	0.07	59
Capital	15	14	12	0.84	0.86	0.80	296
DateOfBirth	15	0	0	0.00	0.00	0.00	15
DateOfDeath	15	1	1	0.07	1.00	0.07	92
Language	15	14	9	0.70	0.64	0.60	167
PlaceOfBirth	15	1	0	0.00	0.00	0.00	125
PlaceOfDeath	15	1	1	0.07	1.00	0.07	163
TOTAL	105	35	24	0.25	0.69	0.23	917
RawNE							
Author	15	1	1	0.07	1.00	0.07	59
Capital	15	13	13	0.87	1.00	0.87	215
DateOfBirth	15	0	0	0.00	0.00	0.00	15
DateOfDeath	15	1	1	0.07	1.00	0.07	92
Language	15	14	9	0.70	0.64	0.60	167
PlaceOfBirth	15	0	0	0.00	0.00	0.00	87
PlaceOfDeath	15	1	1	0.07	1.00	0.07	109
TOTAL	105	30	25	0.25	0.83	0.24	744
Stemmed							
Author	15	3	1	0.07	0.33	0.07	36
Capital	15	15	14	0.93	0.93	0.93	232
DateOfBirth	15	0	0	0.00	0.00	0.00	13
DateOfDeath	15	2	1	0.07	0.50	0.07	79
Language	15	13	9	0.69	0.69	0.60	158
PlaceOfBirth	15	2	0	0.00	0.00	0.00	95
PlaceOfDeath	15	2	1	0.10	0.50	0.07	129
TOTAL	105	37	26	0.26	0.70	0.25	742
StemmedNE							
Author	15	1	1	0.07	1.00	0.07	36
Capital	15	12	12	0.80	1.00	0.80	182
DateOfBirth	15	0	0	0.00	0.00	0.00	13

DateOfDeath	15	1	1	0.07	1.00	0.07	79
Language	15	13	9	0.69	0.69	0.60	158
PlaceOfBirth	15	0	0	0.00	0.00	0.00	67
PlaceOfDeath	15	1	1	0.07	1.00	0.07	85
TOTAL	105	28	24	0.24	0.86	0.23	620
NETagged							
Author	15	11	10	0.68	0.91	0.67	54
Capital	15	13	13	0.87	1.00	0.87	243
DateOfBirth	15	9	8	0.56	0.89	0.53	21
DateOfDeath	15	1	1	0.07	1.00	0.07	69
Language	15	0	0	0.00	0.00	0.00	0
PlaceOfBirth	15	7	7	0.47	1.00	0.47	68
PlaceOfDeath	15	1	1	0.07	1.00	0.07	87
TOTAL	105	42	40	0.39	0.95	0.38	542
AllMethodsWithoutNE							
Author	15	13	9	0.64	0.69	0.60	149
Capital	15	15	14	0.94	0.93	0.93	771
DateOfBirth	15	9	8	0.56	0.89	0.53	49
DateOfDeath	15	2	1	0.07	0.50	0.07	240
Language	15	14	9	0.66	0.64	0.60	325
PlaceOfBirth	15	9	7	0.47	0.78	0.47	288
PlaceOfDeath	15	2	1	0.10	0.50	0.07	379
TOTAL	105	64	49	0.49	0.77	0.47	2201
AllMethodsWithNE							
Author	15	11	10	0.68	0.91	0.67	149
Capital	15	13	13	0.87	1.00	0.87	640
DateOfBirth	15	9	8	0.56	0.89	0.53	49
DateOfDeath	15	1	1	0.07	1.00	0.07	240
Language	15	14	9	0.66	0.64	0.60	325
PlaceOfBirth	15	7	7	0.47	1.00	0.47	222
PlaceOfDeath	15	2	2	0.13	1.00	0.13	281

TOTAL	105	57	50	0.49	0.88	0.48	1906
Threshold: 0.85							
Raw							
Author	15	2	0	0.00	0.00	0.00	25
Capital	15	14	12	0.83	0.86	0.80	118
DateOfBirth	15	0	0	0.00	0.00	0.00	7
DateOfDeath	15	0	0	0.00	0.00	0.00	49
Language	15	7	5	0.37	0.71	0.33	47
PlaceOfBirth	15	1	0	0.00	0.00	0.00	50
PlaceOfDeath	15	1	1	0.07	1.00	0.07	66
TOTAL	105	25	18	0.18	0.72	0.17	362
RawNE							
Author	15	0	0	0.00	0.00	0.00	25
Capital	15	12	12	0.80	1.00	0.80	81
DateOfBirth	15	0	0	0.00	0.00	0.00	7
DateOfDeath	15	0	0	0.00	0.00	0.00	49
Language	15	7	5	0.37	0.71	0.33	47
PlaceOfBirth	15	0	0	0.00	0.00	0.00	37
PlaceOfDeath	15	1	1	0.07	1.00	0.07	44
TOTAL	105	20	18	0.18	0.90	0.17	290
Stemmed							
Author	15	2	0	0.00	0.00	0.00	12
Capital	15	15	14	0.93	0.93	0.93	87
DateOfBirth	15	0	0	0.00	0.00	0.00	6
DateOfDeath	15	1	0	0.00	0.00	0.00	39
Language	15	8	5	0.37	0.63	0.33	50
PlaceOfBirth	15	0	0	0.00	0.00	0.00	35
PlaceOfDeath	15	1	1	0.07	1.00	0.07	53
TOTAL	105	27	20	0.20	0.74	0.19	282
StemmedNE							
Author	15	0	0	0.00	0.00	0.00	12

Capital	15	12	12	0.80	1.00	0.80	69
DateOfBirth	15	0	0	0.00	0.00	0.00	6
DateOfDeath	15	0	0	0.00	0.00	0.00	39
Language	15	8	5	0.37	0.63	0.33	50
PlaceOfBirth	15	0	0	0.00	0.00	0.00	24
PlaceOfDeath	15	0	0	0.00	0.00	0.00	33
TOTAL	105	20	17	0.17	0.85	0.16	233
NETagged							
Author	15	8	7	0.47	0.88	0.47	23
Capital	15	13	13	0.87	1.00	0.87	92
DateOfBirth	15	9	8	0.56	0.89	0.53	13
DateOfDeath	15	0	0	0.00	0.00	0.00	37
Language	15	0	0	0.00	0.00	0.00	0
PlaceOfBirth	15	6	6	0.40	1.00	0.40	35
PlaceOfDeath	15	1	1	0.07	1.00	0.07	37
TOTAL	105	37	35	0.34	0.95	0.33	237
AllMethodsWithoutNE							
Author	15	11	8	0.53	0.73	0.53	60
Capital	15	15	14	0.94	0.93	0.93	297
DateOfBirth	15	9	8	0.56	0.89	0.53	26
DateOfDeath	15	1	0	0.00	0.00	0.00	125
Language	15	12	9	0.63	0.75	0.60	97
PlaceOfBirth	15	8	6	0.40	0.75	0.40	120
PlaceOfDeath	15	1	1	0.07	1.00	0.07	156
TOTAL	105	57	46	0.45	0.81	0.44	881
AllMethodsWithNE							
Author	15	8	7	0.47	0.88	0.47	60
Capital	15	13	13	0.87	1.00	0.87	242
DateOfBirth	15	9	8	0.56	0.89	0.53	26
DateOfDeath	15	0	0	0.00	0.00	0.00	125
Language	15	12	9	0.63	0.75	0.60	97

PlaceOfBirth	15	6	6	0.40	1.00	0.40	96
PlaceOfDeath	15	1	1	0.07	1.00	0.07	114
TOTAL	105	49	44	0.43	0.90	0.42	760
Threshold: 0.9							
Raw							
Author	15	0	0	0.00	0.00	0.00	8
Capital	15	14	12	0.83	0.86	0.80	57
DateOfBirth	15	0	0	0.00	0.00	0.00	0
DateOfDeath	15	0	0	0.00	0.00	0.00	26
Language	15	5	4	0.27	0.80	0.27	16
PlaceOfBirth	15	1	0	0.00	0.00	0.00	22
PlaceOfDeath	15	1	1	0.07	1.00	0.07	18
TOTAL	105	21	17	0.17	0.81	0.16	147
RawNE							
Author	15	0	0	0.00	0.00	0.00	8
Capital	15	12	12	0.80	1.00	0.80	36
DateOfBirth	15	0	0	0.00	0.00	0.00	0
DateOfDeath	15	0	0	0.00	0.00	0.00	26
Language	15	5	4	0.27	0.80	0.27	16
PlaceOfBirth	15	0	0	0.00	0.00	0.00	12
PlaceOfDeath	15	0	0	0.00	0.00	0.00	7
TOTAL	105	17	16	0.15	0.94	0.15	105
Stemmed							
Author	15	0	0	0.00	0.00	0.00	2
Capital	15	11	11	0.73	1.00	0.73	40
DateOfBirth	15	0	0	0.00	0.00	0.00	0
DateOfDeath	15	0	0	0.00	0.00	0.00	22
Language	15	5	4	0.27	0.80	0.27	15
PlaceOfBirth	15	0	0	0.00	0.00	0.00	14
PlaceOfDeath	15	1	1	0.07	1.00	0.07	12
TOTAL	105	17	16	0.15	0.94	0.15	105

StemmedNE							
Author	15	0	0	0.00	0.00	0.00	2
Capital	15	11	11	0.73	1.00	0.73	31
DateOfBirth	15	0	0	0.00	0.00	0.00	0
DateOfDeath	15	0	0	0.00	0.00	0.00	22
Language	15	5	4	0.27	0.80	0.27	15
PlaceOfBirth	15	0	0	0.00	0.00	0.00	7
PlaceOfDeath	15	0	0	0.00	0.00	0.00	2
TOTAL	105	16	15	0.14	0.94	0.14	79
NETagged							
Author	15	3	3	0.20	1.00	0.20	10
Capital	15	12	12	0.80	1.00	0.80	48
DateOfBirth	15	9	8	0.56	0.89	0.53	7
DateOfDeath	15	0	0	0.00	0.00	0.00	18
Language	15	0	0	0.00	0.00	0.00	0
PlaceOfBirth	15	4	4	0.27	1.00	0.27	18
PlaceOfDeath	15	0	0	0.00	0.00	0.00	5
TOTAL	105	28	27	0.26	0.96	0.26	106
AllMethodsWithoutNE							
Author	15	3	3	0.20	1.00	0.20	20
Capital	15	14	13	0.87	0.93	0.87	145
DateOfBirth	15	9	8	0.56	0.89	0.53	7
DateOfDeath	15	0	0	0.00	0.00	0.00	66
Language	15	10	9	0.62	0.90	0.60	31
PlaceOfBirth	15	6	4	0.27	0.67	0.27	54
PlaceOfDeath	15	1	1	0.07	1.00	0.07	35
TOTAL	105	43	38	0.37	0.88	0.36	358
AllMethodsWithNE							
Author	15	3	3	0.20	1.00	0.20	20
Capital	15	12	12	0.80	1.00	0.80	115
DateOfBirth	15	9	8	0.56	0.89	0.53	7

DateOfDeath	15	0	0	0.00	0.00	0.00	66
Language	15	10	9	0.62	0.90	0.60	31
PlaceOfBirth	15	4	4	0.27	1.00	0.27	37
PlaceOfDeath	15	0	0	0.00	0.00	0.00	14
TOTAL	105	38	36	0.35	0.95	0.34	290
Threshold: 0.95							
Raw							
Author	15	0	0	0.00	0.00	0.00	0
Capital	15	13	11	0.77	0.85	0.73	17
DateOfBirth	15	0	0	0.00	0.00	0.00	0
DateOfDeath	15	0	0	0.00	0.00	0.00	4
Language	15	0	0	0.00	0.00	0.00	2
PlaceOfBirth	15	0	0	0.00	0.00	0.00	5
PlaceOfDeath	15	0	0	0.00	0.00	0.00	1
TOTAL	105	13	11	0.11	0.85	0.10	29
RawNE							
Author	15	0	0	0.00	0.00	0.00	0
Capital	15	8	8	0.53	1.00	0.53	8
DateOfBirth	15	0	0	0.00	0.00	0.00	0
DateOfDeath	15	0	0	0.00	0.00	0.00	4
Language	15	0	0	0.00	0.00	0.00	2
PlaceOfBirth	15	0	0	0.00	0.00	0.00	4
PlaceOfDeath	15	0	0	0.00	0.00	0.00	0
TOTAL	105	8	8	0.08	1.00	0.08	18
Stemmed							
Author	15	0	0	0.00	0.00	0.00	0
Capital	15	9	9	0.60	1.00	0.60	12
DateOfBirth	15	0	0	0.00	0.00	0.00	0
DateOfDeath	15	0	0	0.00	0.00	0.00	4
Language	15	0	0	0.00	0.00	0.00	2
PlaceOfBirth	15	0	0	0.00	0.00	0.00	3

PlaceOfDeath	15	0	0	0.00	0.00	0.00	1
TOTAL	105	9	9	0.09	1.00	0.09	22
StemmedNE							
Author	15	0	0	0.00	0.00	0.00	0
Capital	15	8	8	0.53	1.00	0.53	9
DateOfBirth	15	0	0	0.00	0.00	0.00	0
DateOfDeath	15	0	0	0.00	0.00	0.00	4
Language	15	0	0	0.00	0.00	0.00	2
PlaceOfBirth	15	0	0	0.00	0.00	0.00	2
PlaceOfDeath	15	0	0	0.00	0.00	0.00	0
TOTAL	105	8	8	0.08	1.00	0.08	17
NETagged							
Author	15	0	0	0.00	0.00	0.00	2
Capital	15	12	12	0.80	1.00	0.80	18
DateOfBirth	15	3	3	0.20	1.00	0.20	3
DateOfDeath	15	0	0	0.00	0.00	0.00	6
Language	15	0	0	0.00	0.00	0.00	0
PlaceOfBirth	15	0	0	0.00	0.00	0.00	3
PlaceOfDeath	15	0	0	0.00	0.00	0.00	0
TOTAL	105	15	15	0.14	1.00	0.14	32
AllMethodsWithoutNE							
Author	15	0	0	0.00	0.00	0.00	2
Capital	15	13	12	0.80	0.92	0.80	47
DateOfBirth	15	3	3	0.20	1.00	0.20	3
DateOfDeath	15	0	0	0.00	0.00	0.00	14
Language	15	0	0	0.00	0.00	0.00	4
PlaceOfBirth	15	1	0	0.00	0.00	0.00	11
PlaceOfDeath	15	0	0	0.00	0.00	0.00	2
TOTAL	105	17	15	0.14	0.88	0.14	83
AllMethodsWithNE							
Author	15	0	0	0.00	0.00	0.00	2

Capital	15	12	12	0.80	1.00	0.80	35
DateOfBirth	15	3	3	0.20	1.00	0.20	3
DateOfDeath	15	0	0	0.00	0.00	0.00	14
Language	15	0	0	0.00	0.00	0.00	4
PlaceOfBirth	15	0	0	0.00	0.00	0.00	9
PlaceOfDeath	15	0	0	0.00	0.00	0.00	0
TOTAL	105	15	15	0.14	1.00	0.14	67
RERANK							
Raw							
Author	15	14	5	0.39	0.36	0.33	360
Capital	15	15	13	0.93	0.87	0.87	3288
DateOfBirth	15	7	3	0.20	0.43	0.20	90
DateOfDeath	15	9	2	0.13	0.22	0.13	390
Language	15	15	11	0.73	0.73	0.73	2164
PlaceOfBirth	15	14	5	0.39	0.36	0.33	716
PlaceOfDeath	15	5	1	0.07	0.20	0.07	861
TOTAL	105	79	40	0.41	0.51	0.38	7869
RawNE							
Author	15	7	7	0.47	1.00	0.47	360
Capital	15	14	14	0.93	1.00	0.93	2654
DateOfBirth	15	3	3	0.20	1.00	0.20	90
DateOfDeath	15	5	2	0.13	0.40	0.13	390
Language	15	15	11	0.73	0.73	0.73	2164
PlaceOfBirth	15	4	3	0.20	0.75	0.20	516
PlaceOfDeath	15	1	1	0.07	1.00	0.07	617
TOTAL	105	49	41	0.39	0.84	0.39	6791
Stemmed							
Author	15	14	3	0.26	0.21	0.20	196
Capital	15	15	14	0.97	0.93	0.93	2695
DateOfBirth	15	7	3	0.20	0.43	0.20	76
DateOfDeath	15	9	1	0.07	0.11	0.07	316

Language	15	15	9	0.66	0.60	0.60	2043
PlaceOfBirth	15	12	1	0.07	0.08	0.07	514
PlaceOfDeath	15	6	2	0.13	0.33	0.13	721
TOTAL	105	78	33	0.33	0.42	0.31	6561
StemmedNE							
Author	15	6	5	0.33	0.83	0.33	196
Capital	15	14	14	0.93	1.00	0.93	2310
DateOfBirth	15	3	3	0.20	1.00	0.20	76
DateOfDeath	15	3	1	0.07	0.33	0.07	316
Language	15	15	9	0.66	0.60	0.60	2043
PlaceOfBirth	15	2	1	0.07	0.50	0.07	395
PlaceOfDeath	15	2	2	0.13	1.00	0.13	538
TOTAL	105	45	35	0.34	0.78	0.33	5874
NETagged							
Author	15	14	9	0.65	0.64	0.60	240
Capital	15	14	14	0.93	1.00	0.93	2375
DateOfBirth	15	11	9	0.60	0.82	0.60	74
DateOfDeath	15	5	5	0.33	1.00	0.33	235
Language	15	0	0	0.00	0.00	0.00	0
PlaceOfBirth	15	14	12	0.83	0.86	0.80	410
PlaceOfDeath	15	7	2	0.13	0.29	0.13	522
TOTAL	105	65	51	0.50	0.78	0.49	3856
AllMethodsWithoutNE							
Author	15	15	9	0.65	0.60	0.60	796
Capital	15	15	14	0.97	0.93	0.93	8358
DateOfBirth	15	13	9	0.60	0.69	0.60	240
DateOfDeath	15	13	6	0.40	0.46	0.40	941
Language	15	15	10	0.70	0.67	0.67	4207
PlaceOfBirth	15	15	9	0.73	0.60	0.60	1640
PlaceOfDeath	15	12	3	0.20	0.25	0.20	2104
TOTAL	105	98	60	0.61	0.61	0.57	18286

AllMethodsWithNE							
Author	15	14	9	0.65	0.64	0.60	796
Capital	15	14	14	0.93	1.00	0.93	7339
DateOfBirth	15	11	9	0.60	0.82	0.60	240
DateOfDeath	15	8	6	0.40	0.75	0.40	941
Language	15	15	10	0.70	0.67	0.67	4207
PlaceOfBirth	15	14	12	0.83	0.86	0.80	1321
PlaceOfDeath	15	8	3	0.20	0.38	0.20	1677
TOTAL	105	84	63	0.62	0.75	0.60	16521

Table B.1: Results of answer pattern extraction methods at different thresholds

Appendix C

Answer Patterns

Automatically learned answer patterns of seven question types are given in this section. These question types are *Author*, *Capital*, *DateOfBirth*, *DateOfDeath*, *Language*, *PlaceOfBirth*, and *PlaceOfDeath*. In the following list, first the confidence factor is given and then the answer pattern is written. The answer patterns under the caption "AnswerPatternNETagged" are learned by Named Entity Tagged String method, the answer patterns under the caption "AnswerPatternRaw" are learned by Raw String method, and the answer patterns under the caption "AnswerPatternStemmed" are learned by Stemmed String method. First 35 answer patterns are given for each question type - answer pattern extraction method.

Author - AnswerPatternNETagged

0.95 - <Q> yazarı : <A>
0.95 - <Q> (özet) <A>
0.93 - <Q> / <A>
0.92 - <A> " <Q>
0.92 - <Q> yazarı : <A> bas
0.92 - <Q> - özet - <A>
0.92 - <A> <Q>
0.90 - <Q> yazarı : <A> kanuni

0.90 - <Q> <A>
0.90 - <Q> <A> imge
0.89 - <Q> / <A> (
0.89 - <Q> kitap özeti (<A>
0.89 - <Q> kitap özeti (<A>)
0.88 - <A> imzalı <Q>
0.88 - ; <A> imzalı <Q>
0.88 - <Q>'dan hareketle <A>
0.88 - <Q> (my name is red) <A>
0.88 - <Q> , <A> <
0.86 - <Q> <A>
0.86 - <Q> - <A> -
0.86 - <Q> (the clown and his daughter) <A>
0.86 - <Q> - <A> sayfa
0.86 - <A> '<Q>
0.83 - <Q> , yazar <A>
0.83 - <A> un <Q>
0.83 - <A> un <Q>
0.83 - nda <A> un <Q>
0.83 - <Q> büyük boy , <A>
0.83 - <Q> büyük boy , <A> ,
0.83 - <A> in ünlü romanından uyarlanan ve NECountryName da izlenme rekoru kıran <Q>
0.83 -) <A> in ünlü romanından uyarlanan ve NECountryName da izlenme rekoru kıran <Q>
0.83 - <Q> , kitap özeti , <A>
0.83 - <Q> romanının özeti <A>
0.83 - <A> <Q>
0.83 - <Q> NECountryName ilk olarak <A>

Author - AnswerPatternRaw
0.95 - <A>'un <Q>
0.94 - <A>'un " <Q>

0.92 - <Q> yazarı : <A> bas
0.91 - <Q> yazarı : <A>
0.91 - <A>un <Q>
0.90 - <A>un en renkli ve en iyimser romanım , dediği <Q>
0.90 - <A>‘in ünlü romanı <Q>
0.90 - <Q> yazarı : <A> kanuni
0.89 - <Q> / <A> (
0.89 - <A> un <Q>
0.89 - <Q> kitap özeti (<A>
0.89 - <Q> kitap özeti (<A>)
0.88 - <A> imzalı <Q>
0.88 - ; <A> imzalı <Q>
0.88 - <Q>’dan hareketle <A>
0.88 - <Q>’dan hareketle <A>’
0.88 - re <A>’un <Q>
0.88 - <Q> (my name is red) <A>
0.88 - — <A>‘in ünlü romanı <Q>
0.86 - <Q> (özet) <A>
0.86 - <A>’un kitapları , en son <Q>
0.86 - <A>’un ’<Q>
0.86 - <A>in ölümsüz eseri <Q>
0.86 - <Q> (the clown and his daughter) <A>
0.86 - <A>’in “ <Q>
0.85 - <Q> - özet - <A>
0.83 - <A>’un ’en renkli ve en iyimser romanım’ , dediği ’<Q>
0.83 - yazar <A>’un “ <Q>
0.83 - nda <A> un <Q>
0.83 - <Q> büyük boy , <A>
0.83 - <Q> büyük boy , <A> ,
0.83 - <A> in ünlü romanından uyarlanan ve rusya da izlenme rekoru kıran <Q>
0.83 -) <A> in ünlü romanından uyarlanan ve rusya da izlenme rekoru kıran <Q>

0.83 - <A> , <Q>
0.83 - <A> en ünlü romanı '<Q>

Author - AnswerPatternStemmed
0.92 - <Q> yaza : <A> bas
0.90 - <Q> yaza : <A> kanu
0.89 - <Q> / <A> (
0.89 - <A> un <Q>
0.89 - <Q> kita özet (<A>
0.89 - <Q> kita özet (<A>)
0.88 - <Q> yaza : <A>
0.88 - <A> imza <Q>
0.88 - ; <A> imza <Q>
0.88 - <Q> hare <A>
0.88 - <Q> (my name is red) <A>
0.86 - <Q> (the clow and his daug) <A>
0.85 - <Q> - özet - <A>
0.83 - kaps <A> un <Q>
0.83 - <Q> büyü boy , <A>
0.83 - <Q> büyü boy , <A> ,
0.83 - <A> in ünlü roma uyar ve rusy da izle reko kıra <Q>
0.83 -) <A> in ünlü roma uyar ve rusy da izle reko kıra <Q>
0.83 - <A> , <Q>
0.83 - <A> en ünlü roma '<Q>
0.83 - <Q> türk ilk olar <A>
0.83 - <Q> [<A> ran
0.83 - <A> - rutk aziz , <Q>
0.83 - - <A> - rutk aziz , <Q>
0.83 - <A> / (<Q>
0.83 - <Q> <<<A>
0.83 - <Q> (özet) <A>
0.82 - <Q> <A>
0.80 - <Q> , damg , duda kalb , göky , kızı dall , <A>

0.80 - <Q> , damg , duda kalb , göky , kızı dall , <A> ,
0.80 - <Q> roma olma üzer eser türk edeb klas imza atan biri olan <A>
0.80 - <A> 'in aynı adlı roma uyar '<Q>
0.80 - <A> , hırs poli , hürr , roma , şems inka , sena gökh akte , <Q>
0.80 - , <A> , hırs poli , hürr , roma , şems inka , sena gökh akte , <Q>
0.80 - <A> in <Q>

Capital - AnswerPatternNETagged

0.99 - <Q>'ın başkenti <A>
0.99 - <Q>'nin başkenti <A>
0.99 - <Q>'nin başkenti <A> .
0.98 - <Q>'nın başkenti <A> ,
0.98 - <Q>'nın başkenti <A>'
0.98 - <Q>'nın başkenti <A> ya
0.97 - <Q>'nın başkenti <A>
0.97 - <Q>'nın başkenti <A> .
0.97 - steaua <A> <Q>
0.97 - <Q>'in başkenti <A>
0.97 - <Q>'nin başkenti <A> ş
0.97 - <Q>'nın başkenti <A> d
0.96 - <Q>'ın başkenti <A>'
0.96 - steaua <A> , <Q>
0.95 - <Q> turu (NEPersonName - <A>
0.95 - <Q> turu (NEPersonName - <A>)
0.95 - <Q>'nın başkenti <A> ve
0.95 - <Q>'ın başkenti <A> yak
0.94 - <Q>'in başkenti <A> ,
0.94 - <Q>'nin başkenti <A> ,
0.94 - <Q>'nın steaua <A>
0.94 - <Q>'nın steaua <A> tak
0.94 - real <A> (<Q>
0.94 - <A> , <Q>
0.94 - <Q>'nin palmyra , NEPersonName , <A>

0.94 - <Q>'nin palmyra , NEPersonName , <A> ,
0.94 - <Q> turu , <A> ,
0.93 - <A> / <Q>
0.93 - <Q>'da başkent <A>
0.93 - <Q> türk havayolları ile (<A>
0.93 - <Q> türk havayolları ile (<A> gidi
0.93 - <Q> türk havayolları ile (venedik gidiş - <A>
0.93 - <Q> türk havayolları ile (venedik gidiş - <A> d
0.92 - <Q> yerleşim yerleri taslakları — <A>
0.91 - real <A> <Q>

Capital - AnswerPatternRaw
0.99 - <Q>'ın başkenti <A>'
0.99 - <Q> turları — mısır turları — ürdün turu — beyrut turu — dubai turları — halep turları — <A>
0.99 - <Q> turları — mısır turları — ürdün turu — beyrut turu — dubai turları — halep turları — <A> turlar
0.96 - <Q>'nın başkenti <A>'
0.96 - <Q>'in başkenti <A>'
0.96 - steaua <A> , <Q>
0.96 - <Q>'nin başkenti <A> .
0.96 - <Q>nın başkenti <A>
0.95 - <Q> şatolar ve <A>
0.95 - <Q> turu (halep - <A>
0.95 - <Q> turu (halep - <A>)
0.95 - <Q>'nın başkenti <A> ,
0.95 - <Q> şatolar ve <A><
0.95 - <Q> şatolar ve <A> ,
0.95 - <Q>'nin başkenti <A>'
0.95 - <Q> cannes<Q> nice<Q> <A>
0.95 - <Q> cannes<Q> nice<Q> <A><
0.95 - <Q>'nin başkenti <A>
0.95 - <Q>nin başkenti <A>

0.94 - <Q>'nin başkenti <A> ş
0.94 - <Q>'in başkenti <A> ,
0.94 - <Q>'nin başkenti <A> ,
0.94 - <Q>'nın steaua <A>
0.94 - <Q>'nın steaua <A> tak
0.94 - real <A> (<Q>
0.94 - <Q>'nın başkenti <A> ya
0.94 - <Q>'nin palmyra , halep , <A>
0.94 - <Q>'nin palmyra , halep , <A> ,
0.94 - <Q>nin başkenti <A>
0.94 - <Q> floransa<Q> <A>
0.94 - <Q> floransa<Q> <A><
0.94 - <Q>'nın başkenti <A> d
0.93 - <Q>'da başkent <A>
0.93 - <Q>'da başkent <A>'
0.93 - <Q> türk havayolları ile (venedik gidiş - <A>

Capital - AnswerPatternStemmed

0.99 - <Q> turl — mısı turl — ürdü turu — beyr turu — duba turl — hale turl — <A>
0.99 - <Q> turl — mısı turl — ürdü turu — beyr turu — duba turl — hale turl — <A> turl
0.98 - <Q> başk <A> ,
0.97 - <Q> — umre — gap turu — <Q> turu — ürdü turl — kara turu — ürdü turu — mısı — hale ve <A>
0.97 - <Q> — umre — gap turu — <Q> turu — ürdü turl — kara turu — ürdü turu — mısı — hale ve <A> gezi
0.97 - <Q> başk <A>'
0.96 - <Q> başk <A>
0.96 - stea <A> , <Q>
0.95 - <Q> turu (hale - <A>
0.95 - <Q> turu (hale - <A>)
0.95 - <Q> <A> loir

0.95 - <Q> cann nice <A>
0.95 - <Q> stea <A>
0.95 - <Q> stea <A> tak
0.94 - <Q> başk <A>
0.94 - real <A> (<Q>
0.94 - <Q> palm , hale , <A>
0.94 - <Q> palm , hale , <A> ,
0.94 - <Q> flor <A>
0.93 - <Q> başk <A> hava
0.93 - stea <A> <Q>
0.93 - <Q> türk hava ile (vene gidi - <A>
0.93 - <Q> türk hava ile (vene gidi - <A> d
0.93 - <Q> başk <A> ve
0.92 - <Q> başk <A> ya
0.92 - <Q> başk <A> ulus
0.92 - <Q> başk <A> “
0.92 - <Q> tems stea <A>
0.92 - <Q> başk <A> yak
0.92 - <Q> kupa ınte , <A>
0.92 - <Q> kupa ınte , <A> ve
0.91 - <Q> savu baka davi siha , başk <A>
0.91 - <Q> turu (hale - <A> -
0.90 - <Q> başk <A>
0.90 - <A> bağd , 1982 <Q>

DateOfBirth - AnswerPatternNETagged

0.98 - <Q> (d . <A>
0.98 - <Q> (d . <A> ,
0.98 - <Q> (d . <A> NEDate
0.93 - <Q> (<A> -
0.93 - <Q> , (d . <A>
0.93 - <Q> , (d . <A> ,
0.91 - <Q> (<A> ,

0.89 - <Q> “ türk sineması emektarı “ (<A>
0.89 - <Q> “ türk sineması emektarı “ (<A> -
0.88 - <Q> , [1] (d . <A>
0.88 - <Q> , [1] (d . <A> ,
0.86 - <Q> , geçen yüzyılın başında <A>
0.86 - <Q> , geçen yüzyılın başında <A>’
0.83 - <Q>’nün hayat biyografisi & konusunu görüntülemektesiniz . NEPerson-Name <Q> (<A>
0.83 - <Q>’nün hayat biyografisi & konusunu görüntülemektesiniz . NEPerson-Name <Q> (<A> -
0.83 - <Q> <A> 27
0.81 - <Q> (<A>
0.80 - <Q> <A>
0.80 - <Q> için her yılın NEDate uğurlu gündü . . . çünkü <A>
0.80 - <Q> , çemberlitaş , NECityName [4] <A>
0.80 - <Q> , çemberlitaş , NECityName [5] <A>
0.78 - <Q> . <A>
0.75 - <A> - 25 NEDate) konusunu görüntülemektesiniz . NEPersonName <Q>
0.75 - (<A> - 25 NEDate) konusunu görüntülemektesiniz . NEPersonName <Q>
0.75 - <Q> aslen , çok uzun geçmişiyle ankaralı olan <Q> , <A>
0.75 - <Q> aslen , ankaralı olan <Q> , <A>
0.75 - <Q> NEPersonName tarihi : <A>
0.75 - <Q> NEPersonName tarihi : <A> -
0.75 - <Q> <Q> , <A>
0.75 - <Q> “ , “ NEPersonName “ , (d . <A>
0.75 - <Q> “ , “ NEPersonName “ , (d . <A> ,
0.75 - <Q> — d <A>
0.75 - <Q> — d <A> NECityName
0.67 - <Q>’ün doğum tarihi <A>
0.67 - <Q>’ün doğum tarihi <A> .

DateOfBirth - AnswerPatternRaw

0.89 - <Q> “ türk sineması emektarı “ (<A>
0.89 - <Q> “ türk sineması emektarı “ (<A> -
0.88 - <Q> . <A> tarihinde
0.88 - <Q> , [1] (d . <A>
0.88 - <Q> , [1] (d . <A> ,
0.86 - <Q> , geçen yüzyılın başında <A>
0.86 - <Q> , geçen yüzyılın başında <A>’
0.83 - <Q>’nün hayat biyografisi & konusunu görüntülemektesiniz . mustafa <Q> (<A>
0.83 - <Q>’nün hayat biyografisi & konusunu görüntülemektesiniz . mustafa <Q> (<A> -
0.80 - <Q> için her yılın 5 nisanı uğurlu gündü . . . çünkü <A>
0.80 - <Q> için her yılın 5 nisanı uğurlu gündü . . . çünkü <A>’
0.80 - <Q> , çemberlitaş , istanbul’da [4] <A>
0.80 - <Q> , çemberlitaş , istanbul’da [4] <A> tarihi
0.80 - <Q> , çemberlitaş , istanbul’da [5] <A>
0.80 - <Q> , çemberlitaş , istanbul’da [5] <A> tarihi
0.75 - <A> - 25 aralık 1973) konusunu görüntülemektesiniz . mustafa <Q>
0.75 - (<A> - 25 aralık 1973) konusunu görüntülemektesiniz . mustafa <Q>
0.75 - <A>’de kırşehir’de doğan <Q>
0.75 - <A> de kırşehirde doğan <Q>
0.75 - <Q> aslen , çok uzun geçmişiyle ankaralı olan <Q> , <A>
0.75 - <Q> aslen , çok uzun geçmişiyle ankaralı olan <Q> , <A> y
0.75 - <Q> aslen , ankaralı olan <Q> , <A>
0.75 - <Q> aslen , ankaralı olan <Q> , <A> y
0.75 - <Q> doğum tarihi : <A>
0.75 - <Q> doğum tarihi : <A> -
0.75 - <Q> <Q> , <A>
0.75 - <Q> <Q> , <A>’
0.75 - <Q> “ , “ mehmet ragif “ , (d . <A>
0.75 - <Q> “ , “ mehmet ragif “ , (d . <A> ,
0.75 - <Q> — d <A> istanbul
0.71 - <Q> (d . <A> kas

0.67 - <Q>'ün doğum tarihi <A>
0.67 - <Q>'ün doğum tarihi <A> .
0.67 - <Q> türkiye cumhuriyeti cumhurbaşkanı doğum tarihi <A>
0.67 - <Q> türkiye cumhuriyeti cumhurbaşkanı doğum tarihi <A> do

DateOfBirth - AnswerPatternStemmed

0.89 - <Q> " türk sine emek " (<A>
0.89 - <Q> " türk sine emek " (<A> -
0.88 - <Q> . <A> tari
0.88 - <Q> , [1] (d . <A>
0.88 - <Q> , [1] (d . <A> ,
0.86 - <Q> , geçe yüzy başı <A>
0.83 - <Q> haya biyo & konu görü . must <Q> (<A>
0.83 - <Q> haya biyo & konu görü . must <Q> (<A> -
0.80 - <Q> için her yılı 5 nisa uğur günd . . . çünk <A>
0.80 - <Q> , çemb , ista [4] <A>
0.80 - <Q> , çemb , ista [4] <A> tari
0.80 - <Q> , çemb , ista [5] <A>
0.80 - <Q> , çemb , ista [5] <A> tari
0.75 - <A> - 25 aral 1973) konu görü . must <Q>
0.75 - (<A> - 25 aral 1973) konu görü . must <Q>
0.75 - <A> de kırş doğa <Q>
0.75 - <Q> asle , çok uzun geçm anka olan <Q> , <A>
0.75 - <Q> asle , çok uzun geçm anka olan <Q> , <A> y
0.75 - <Q> asle , anka olan <Q> , <A>
0.75 - <Q> asle , anka olan <Q> , <A> y
0.75 - <Q> doğu tari : <A>
0.75 - <Q> doğu tari : <A> -
0.75 - <Q> <Q> , <A>
0.75 - <Q> " , " mehm ragi " , (d . <A>
0.75 - <Q> " , " mehm ragi " , (d . <A> ,
0.75 - <Q> — d <A> ista
0.71 - <Q> (d . <A> kas

0.67 - <Q> doğu tari <A>
0.67 - <Q> doğu tari <A> .
0.67 - <Q> türk cumh cumh doğu tari <A>
0.67 - <Q> türk cumh cumh doğu tari <A> do
0.67 - <A> - 25 aral 1973) konu görü . = > must <Q>
0.67 - (<A> - 25 aral 1973) konu görü . = > must <Q>
0.67 - <Q> , ülke buna bir döne , <A>
0.67 - <Q> , ülke buna bir döne , <A> tari

DateOfDeath - AnswerPatternNETagged

0.96 - <Q> ise , beş günlük yoğun bakımın ardından <A>
0.96 - <Q> ise , beş günlük yoğun bakımın ardından <A> ç
0.95 - <Q> (d . 29 NEDate , NECityName ö . <A>
0.95 - <Q> (d . 29 NEDate , NECityName ö . <A> ,
0.95 - <Q> (d . NEDate , NECityName - ö . <A>
0.95 - <Q> (d . NEDate , NECityName - ö . <A> ,
0.95 - <Q> , (1 NEDate , NECityName - <A>
0.95 - <Q> , (1 NEDate , NECityName - <A> ,
0.94 - <Q> - NEPersonName ölmez - <A>
0.92 - <Q>ın vefatı üzerine <A>
0.92 - <Q>ın vefatı üzerine <A> -
0.92 - <Q> yazılarını yazmaya devam ederken uzun süren bir hastalık dönemi geçirdi ve sonra <A>
0.92 - <Q> , bürokratik oligarşi kurbanı (<A>
0.92 - <Q> , bürokratik oligarşi kurbanı (<A> ç
0.92 - <Q> , (d . NEDate NECityName ö . <A>
0.92 - <Q> , (d . NEDate NECityName ö . <A> NECityName
0.91 - <Q> , bitkisel hayata girdikten NEQuantity sonra <A>
0.90 - <Q> , (d . NEDate NECityName ö . <A> ,
0.89 - <Q> , bitkisel hayata girdikten NEQuantity sonra <A> pazar
0.88 - <Q> (1881 - <A>
0.88 - <Q> (1881 - <A>)
0.88 - <Q>'ın vefatı üzerine <A>

0.88 - <Q>'ın vefatı üzerine <A> -
0.88 - <Q> (NECityName , 25 NEDate - NECityName , <A>
0.88 - <Q> (NECityName , 25 NEDate - NECityName , <A>)
0.88 - <Q> , 1 NEDate NELocationName doğdu , <A>
0.88 - <Q> , (d . 29 NEDate , NECityName ö . <A>
0.88 - <Q> , (d . 29 NEDate , NECityName ö . <A> ,
0.88 - <Q> , NEPersonName inan ve NEPersonName birlikte <A>
0.88 - <Q> , NEPersonName inan ve NEPersonName birlikte <A> sabah
0.87 - <Q> (NEDate - <A>
0.87 - <Q> (NEDate - <A>)
0.86 - <Q> , (1 NEDate , urfa - <A>
0.86 - <Q> , (1 NEDate , urfa - <A> ,
0.86 - <Q> , NEPersonName ve NEPersonName inanın , <A>

DateOfDeath - AnswerPatternRaw

0.96 - <Q> ise , beş günlük yoğun bakımın ardından <A>
0.96 - <Q> ise , beş günlük yoğun bakımın ardından <A> ç
0.95 - <Q> (d . 29 haziran 1979 , zonguldak ö . <A>
0.95 - <Q> (d . 29 haziran 1979 , zonguldak ö . <A> ,
0.95 - <Q> , (1 nisan 1937 , adana - <A>
0.95 - <Q> , (1 nisan 1937 , adana - <A> ,
0.94 - <A> günü dönemin cumhurbaşkanı <Q>
0.94 - <A>'te yaşamını yitiren araştırmacı - gazeteci <Q>
0.94 - sonucu <A>'te yaşamını yitiren araştırmacı - gazeteci <Q>
0.92 - <A>'te , yaşamını yitiren araştırmacı gazeteci <Q>
0.92 - , <A>'te , yaşamını yitiren araştırmacı gazeteci <Q>
0.92 - <Q>ın vefatı üzerine <A>
0.92 - <Q>ın vefatı üzerine <A> -
0.92 - <Q> yazılarını yazmaya devam ederken uzun süren bir hastalık dönemi geçirdi ve sonra <A>
0.92 - <Q> yazılarını yazmaya devam ederken uzun süren bir hastalık dönemi geçirdi ve sonra <A>'
0.92 - <Q> , bürokratik oligarşi kurbanı (<A>

0.92 - <Q> , bürokratik oligarşi kurbanı (<A> ç
0.92 - <Q> , (d . 28 mayıs 1925 istanbul ö . <A>
0.92 - <Q> , (d . 28 mayıs 1925 istanbul ö . <A> ankara
0.91 - <Q> (d . 5 nisan 1925 , istanbul - ö . <A>
0.91 - <Q> (d . 5 nisan 1925 , istanbul - ö . <A> ,
0.91 - <A> tarihinde istanbul'da ölen sanatçının anısına , eşi çolpan ilhan tarafından kurulan <Q>
0.91 - <Q> , bitkisel hayata girdikten 172 gün sonra <A>
0.90 - <Q> (5 nisan 1925 - <A>
0.90 - <Q> (5 nisan 1925 - <A>)
0.90 - <Q> , (d . 28 mayıs 1925 istanbul ö . <A> ,
0.89 - <Q> , bitkisel hayata girdikten 172 gün sonra <A> pazar
0.89 - <Q> ve arkadaşlarının <A> tarihinde
0.88 - <Q> - uğurlar ölmez - <A>
0.88 - <A>'de hayata gözlerini yuman türkiye cumhuriyeti'nin kurucusu büyük önder <Q>
0.88 - <Q>'ın vefatı üzerine <A>
0.88 - <Q>'ın vefatı üzerine <A> -
0.88 - <Q> (istanbul , 25 kasım 1889 - londra , <A>
0.88 - <Q> (istanbul , 25 kasım 1889 - londra , <A>)
0.88 - <A> tarihi , türkiyenin ender yetiştirdiği komünist sanatçılardan birisi olan <Q>

DateOfDeath - AnswerPatternStemmed

0.96 - <Q> ise , beş günl yoğu bakı ardı <A>
0.96 - <Q> ise , beş günl yoğu bakı ardı <A> ç
0.95 - <Q> (d . 29 hazi 1979 , zong ö . <A>
0.95 - <Q> (d . 29 hazi 1979 , zong ö . <A> ,
0.95 - <Q> , (1 nisa 1937 , adan - <A>
0.95 - <Q> , (1 nisa 1937 , adan - <A> ,
0.94 - <A> günü döne cumh <Q>
0.93 - <A> tari ista ölen sana anıs , eşi çolp ilha tara kuru <Q>
0.92 - <Q> vefa üzer <A> -

0.92 - <Q> yazı yazm deva eder uzun süre bir hast döne geçi ve sonr <A>
0.92 - <Q> , büro olig kurb (<A>
0.92 - <Q> , büro olig kurb (<A> ç
0.92 - <Q> , (d . 28 mayı 1925 ista ö . <A>
0.92 - <Q> , (d . 28 mayı 1925 ista ö . <A> anka
0.91 - <Q> (d . 5 nisa 1925 , ista - ö . <A>
0.91 - <Q> (d . 5 nisa 1925 , ista - ö . <A> ,
0.91 - <Q> , bitk haya gird 172 gün sonr <A>
0.90 - <Q> (5 nisa 1925 - <A>
0.90 - <Q> (5 nisa 1925 - <A>)
0.90 - <Q> , (d . 28 mayı 1925 ista ö . <A> ,
0.90 - <A> tari uluc ceza idam edil 37'n yıld , <Q>
0.90 - inan <A> tari uluc ceza idam edil 37'n yıld , <Q>
0.89 - <Q> , bitk haya gird 172 gün sonr <A> paza
0.88 - <Q> - uğur ölme - <A>
0.88 - <Q> vefa üzer <A>
0.88 - <Q> vefa üzer <A> -
0.88 - <Q> " uzun süre , faka fikr ve yazı yazm enge bir hast sonr eren evin ölmü (<A>
0.88 - <Q> " uzun süre , faka fikr ve yazı yazm enge bir hast sonr eren evin ölmü (<A>)
0.88 - <Q> (ista , 25 kası 1889 - lond , <A>
0.88 - <Q> (ista , 25 kası 1889 - lond , <A>)
0.88 - <A> tari , türk ende yeti komü sana biri olan <Q>
0.88 - <Q> , (d . 29 hazi 1979 , zong ö . <A>
0.88 - <Q> , (d . 29 hazi 1979 , zong ö . <A> ,
0.88 - <Q> , hüse inan ve yusu asla birl <A>
0.88 - <Q> , hüse inan ve yusu asla birl <A> saba

Language - AnswerPatternNETagged

0.33 - <Q> <A>sindeki
0.33 - - <A> : <Q>
0.33 - <Q>'de <A> i

0.33 - — <A> dil eğitimi — <Q>
0.33 - <Q> > > <A>
0.33 - <Q>'da <A> —
0.33 - — <A> kursu — <Q>
0.33 - <Q>ya <A>
0.33 - <Q>ya <A> e
0.33 - <Q> ve <A> hakk
0.33 - <Q> , NECityName <A>
0.33 - <Q> > <A>
0.33 - <A> , ırak , <Q>
0.33 - <Q>da <A> ve
0.33 - <Q>da <A> :
0.33 - <Q> (<A> ad
0.33 - <A> , ırak , NECountryName , <Q>
0.33 - <Q>'da <A> yabanc
0.25 - <Q> - <A> [
0.25 - <Q>de <A>
0.25 - <Q> - <A> NELocationName
0.25 - <Q>'ye <A>
0.25 - <Q>da <A>
0.25 - <Q> da <A> ö
0.25 - <A> ; <Q>
0.25 - <Q> NELocationName <A>
0.25 - <Q> tarihi <A>
0.25 - <Q>da , <A>
0.25 - <Q> : <A> t
0.20 - <Q>'da <A> dil
0.17 - (<A> - <Q>
0.17 - <Q> da <A> e
0.17 - <Q> da <A>
0.17 - <Q> da <A>n
0.17 - <Q>da <A> ve

Language - AnswerPatternRaw

0.95 - <A> : rpublique française) ya da kısaca <Q>

0.95 - (<A> : rpublique française) ya da kısaca <Q>

0.94 - <A> kursu — <Q>

0.93 - da <A> kursu — <Q>

0.92 - <Q> federasyonu (<A> :

0.92 - <A> : , rossiyskaya federatsiya) , kısaca <Q>

0.92 - (<A> : , rossiyskaya federatsiya) , kısaca <Q>

0.92 - <Q> arap cumhuriyeti (<A>

0.92 - <Q> haritası + <A>

0.92 - <Q> haritası + <A> yama

0.92 - da <A> dil okulları - egitim<Q>

0.92 - e <A> sözlük , turkish russian dictionary , russian turkish dictionary , dil , eğitim , <Q>

0.91 - <Q> cumhuriyeti (<A> :

0.90 - <Q> arap cumhuriyeti (<A> :

0.90 - <Q>'ın ve ermeni diasporasının terminolojisini ve yorumunu <A>

0.90 - <Q>'ın ve ermeni diasporasının terminolojisini ve yorumunu <A> payla

0.89 - <A> :) ya da kısaca <Q>

0.89 - (<A> :) ya da kısaca <Q>

0.89 - <Q>nın iki resmi dili vardır , <A>

0.89 - <Q>nın iki resmi dili vardır , <A> ve

0.89 - <Q>lılar tarafından çoğu kez ülkenin ismi olan <A>

0.89 - <Q>lılar tarafından çoğu kez ülkenin ismi olan <A> misru

0.89 - <A> misru , <Q>

0.89 - olan <A> misru , <Q>

0.89 - <Q> federal cumhuriyeti ya da kısaca <Q> (<A>

0.89 - <Q> federal cumhuriyeti ya da kısaca <Q> (<A> :

0.88 - <Q> dışındaki <A>

0.88 - <Q> dışındaki <A> edebiyat

0.88 - <Q> , <Q>da eğitim ve <Q>da <A>

0.88 - <Q> , <Q>da eğitim ve <Q>da <A> dil

0.88 - <Q> parlamentosu (<A>

0.88 - <Q> parlamentosu (<A>)
0.88 - <Q> , resmiy adı <Q> cumhuriyeti (<A>
0.88 - <Q> , resmi adıyla <Q> cumhuriyeti (<A>
0.88 - <Q> , resmi adıyla <Q> cumhuriyeti (<A> :

Language - AnswerPatternStemmed
0.95 - <A> : rpu fran) ya da kısa <Q>
0.95 - (<A> : rpu fran) ya da kısa <Q>
0.93 - ' <A> kurs — <Q>
0.92 - <Q> fede (<A> :
0.92 - <A> : , ross fede) , kısa <Q>
0.92 - (<A> : , ross fede) , kısa <Q>
0.92 - <Q> arap cumh (<A>
0.92 - <Q> hari + <A>
0.92 - <Q> hari + <A> yama
0.92 - rk <A> sözl , turk russ dict , russ turk dict , dil , eğit , <Q>
0.92 - ' <A> dil eğit — <Q>
0.91 - <Q> cumh (<A> :
0.90 - <Q> arap cumh (<A> :
0.90 - <Q> ve erme dias term ve yoru <A>
0.90 - <Q> ve erme dias term ve yoru <A> payl
0.89 - <A> kurs — <Q>
0.89 - <A> :) ya da kısa <Q>
0.89 - (<A> :) ya da kısa <Q>
0.89 - <Q> iki resm dili vard , <A>
0.89 - <Q> iki resm dili vard , <A> ve
0.89 - <Q> tara çoğu kez ülke ismi olan <A>
0.89 - <Q> tara çoğu kez ülke ismi olan <A> misr
0.89 - <A> misr , <Q>
0.89 - olan <A> misr , <Q>
0.89 - <Q> fede cumh ya da kısa <Q> (<A>
0.89 - <Q> fede cumh ya da kısa <Q> (<A> :
0.88 - <Q> dışı <A> edeb

0.88 - <Q> , <Q>d eğit ve <Q>d <A>
0.88 - <Q> , <Q>d eğit ve <Q>d <A> dil
0.88 - <A> dil okul , <Q>
0.88 - <Q> parl (<A>
0.88 - <Q> parl (<A>)
0.88 - <A> düny , çoğu <Q>
0.88 - mode <A> düny , çoğu <Q>
0.88 - <Q> , resm adı <Q> cumh (<A>

PlaceOfBirth - AnswerPatternNETagged

0.96 - <Q> , (d . 25 NEDate , <A>
0.95 - <Q> , (d . 25 NEDate , <A> ,
0.95 - <Q> - NEPersonName <A>
0.95 - <Q> ve <A> konulu
0.94 - <Q> - NEPersonName <A> ş
0.92 - <Q> NEPersonName <A>
0.92 - <Q> NEPersonName <A> video
0.90 - <Q> (d . NEDate , <A>
0.90 - <Q> (d . NEDate , <A>)
0.90 - <Q> , 1881 (rumi 1296) yılında <A>
0.90 - <Q> sanat merkezi - alsancak - <A>
0.90 - <Q> , <A> valisi
0.90 - <Q> 26 NEDate <A>
0.90 - <Q> (d . 26 NEDate , <A>
0.90 - <Q> <A> milletvekili
0.90 - <Q> , memleketi <A>
0.90 - <Q> programda , yıllar önce ödül aldığı “ <A>
0.90 - <Q> programda , yıllar önce ödül aldığı “ <A> ekspresi
0.89 - <Q> (d . NEDate , <A> ,
0.88 - <Q> NEDate <A>
0.88 - <Q>’ün anne soyu da , NECityName / NECityName gelerek <A>
0.88 - <Q>’ün anne soyu da , NECityName / NECityName gelerek <A> ile
0.88 - <Q>’ün kız kardeşi olan NEPersonName , NEDate <A>

0.88 - <Q>'ın davetlisi olarak <A>
0.88 - <Q> , NEDate NELocationName başkanı , NEDate ise refah partisi <A>
0.88 - <Q> , NEDate NELocationName başkanı , NEDate ise refah partisi <A> il
0.88 - <Q> , tedavi gördüğü başkent hastanesi <A>
0.88 - <Q> , tedavi gördüğü başkent hastanesi <A> sa
0.86 - <Q> (d . NEDate , <A> -
0.86 - <Q> ölüm yıldönümü münasebetiyle memleketi <A>
0.86 - <Q> ölüm yıldönümü münasebetiyle memleketi <A>'
0.86 - <Q> tiyatrosu sakıp sabancı müzesi sakman club salsanat <A>
0.86 - <Q> tiyatrosu sakıp sabancı müzesi sakman club salsanat <A> NELocationName
0.86 - NEPersonName <A> (<Q>
0.86 - <Q> , <A> pendik

PlaceOfBirth - AnswerPatternRaw

0.98 - <A>'da vefat eden " türk şiirinin büyük şairi " <Q>
0.98 - de <A>'da vefat eden " türk şiirinin büyük şairi " <Q>
0.96 - <Q> , (d . 25 şubat 1935 , <A>
0.95 - <Q> - canım <A>
0.95 - <Q> , (d . 25 şubat 1935 , <A> ,
0.94 - <Q> - canım <A> ş
0.93 - <Q> canım <A>
0.92 - <Q> , 22 ağustos 1942 yılında , babasının memuriyeti dolayısıyla <A>
0.92 - <Q> canım <A> video
0.92 - <Q> , memleketi <A>
0.92 - <Q> , 22 ağustos 1942 yılında , babasının memuriyeti dolayısıyla <A>'
0.91 - <Q> ve <A> konulu
0.91 - <Q> , 1881 (rumi 1296) yılında <A>
0.91 - <Q> , 1884 yılında <A>
0.91 - <Q> , memleketi <A>'
0.90 - <Q> , 1881 (rumi 1296) yılında <A>'
0.90 - <Q> sanat merkezi - alsancak - <A>

0.90 - <Q> 5 mart 1925 yılında <A>
0.90 - <Q> 26 şubat 1954'te <A>
0.90 - <Q> 26 şubat 1954'te <A>'
0.90 - <Q> programda , yıllar önce ödül aldığı " <A>
0.90 - <Q> programda , yıllar önce ödül aldığı " <A> ekspresi
0.89 - <Q> , 4 nisan 1967'de <A>
0.89 - <Q> , 4 nisan 1967'de <A>'
0.88 - <Q>'ün anne soyu da , konya / karaman'dan gelerek <A>
0.88 - <Q>'ün anne soyu da , konya / karaman'dan gelerek <A> ile
0.88 - <Q>'ün kız kardeşi olan makbule atadan , 1887 yılında <A>
0.88 - <Q>'ün kız kardeşi olan makbule atadan , 1887 yılında <A>'
0.88 - <Q> , 19 mayıs 1881 yılında , <A>
0.88 - <A>'da dünyaya gelen <Q>
0.88 - nda <A>'da dünyaya gelen <Q>
0.88 - <Q> 5 mart 1925 yılında <A>'
0.88 - <Q>'ın davetlisi olarak <A>
0.88 - <Q>'ın davetlisi olarak <A>'
0.88 - <Q> , 1984 yılında refah partisi beyoğlu ilçe başkanı , 1985 yılında ise refah partisi <A>

PlaceOfBirth - AnswerPatternStemmed
0.96 - <Q> , (d . 25 şuba 1935 , <A>
0.95 - <Q> - canı <A>
0.95 - <Q> , (d . 25 şuba 1935 , <A> ,
0.94 - <Q> - canı <A> ş
0.93 - <Q> canı <A>
0.92 - <Q> , 22 ağus 1942 yılı , baba memu dola <A>
0.92 - <Q> canı <A> vide
0.91 - <Q> , 1881 (rumi 1296) yılı <A>
0.91 - <Q> , 1884 yılı <A>
0.90 - <Q> sana merk - alsa - <A>
0.90 - <Q> 5 mart 1925 yılı <A>
0.90 - <Q> 26 şuba 1954 <A>

0.90 - <Q> prog , yıll önce ödül aldı “ <A>
0.90 - <Q> prog , yıll önce ödül aldı “ <A> eksp
0.89 - <Q> , 4 nisa 1967 <A>
0.88 - <Q> anne soyu da , kony / kara gele <A>
0.88 - <Q> anne soyu da , kony / kara gele <A> ile
0.88 - <Q> kız kard olan makb atad , 1887 yılı <A>
0.88 - <Q> , 19 mayı 1881 yılı , <A>
0.88 - <Q> dave olar <A>
0.88 - <Q> , 1984 yılı refa part beyo ilçe başk , 1985 yılı ise refa part <A>
0.88 - <Q> , 1984 yılı refa part beyo ilçe başk , 1985 yılı ise refa part <A> il
0.88 - <Q> canı <A>
0.88 - <Q> canı <A> ş
0.88 - <Q> , teda görd başk hast <A>
0.88 - <Q> , teda görd başk hast <A> sa
0.86 - <Q> ölüm yıld müna meml <A>
0.86 - <Q> ölüm yıld müna meml <A>’
0.86 - <Q> , dern , <A>
0.86 - <Q> , dern , <A> ,
0.86 - <Q> tiya sakı saba müze sakm club sals <A>
0.86 - <Q> tiya sakı saba müze sakm club sals <A> sama
0.86 - <Q> , 11 kası 1944 tari türk <A>
0.86 - <Q> , 11 kası 1944 tari türk <A> ilin
0.86 - <A> bele başk <Q>

PlaceOfDeath - AnswerPatternNETagged

0.92 - <Q> NEPersonName <A>
0.92 - <Q> NEPersonName <A> video
0.92 - <Q> sanki hala <A>
0.92 - <A> hastanesi’ne kaldırılan <Q>
0.92 - zel <A> hastanesi’ne kaldırılan <Q>
0.89 - <Q> , hastalanması üzerine tedavi için gönderildiği <A>
0.88 - <Q> belgeseli “ galası NEDate <A>
0.88 - <A> gop <Q>

0.88 - <A> gop <Q>
0.88 - <Q> NEDate mülteci olarak yaşadığı <A>
0.88 - <Q>'nun ölümünden sonra <A>
0.88 - <Q> , NEPersonName inan ve NEPersonName , idam edilişlerinin 37 . yılında <A>
0.88 - <Q> , NEPersonName inan ve NEPersonName , idam edilişlerinin 37 . yılında <A> kar
0.88 - <Q> , NEPersonName ve NEPersonName inan idam edilişlerinin 37 . yıldönümünde <A>
0.88 - <Q> , NEPersonName ve NEPersonName inan idam edilişlerinin 37 . yıldönümünde <A> kar
0.86 - <Q> NELocationName <A> asfalt
0.86 - <Q> cinayetinden önce <A>
0.86 - <Q> tiyatrosu sakıp sabancı müzesi sakman club salsanat <A>
0.86 - <Q> tiyatrosu sakıp sabancı müzesi sakman club salsanat <A> NELocationName
0.86 - NEPersonName <A> (<Q>
0.86 - <Q> , evliliğinde de <A>
0.86 - <Q> , evliliğinde de <A> gelene
0.86 - <Q> , yanındaki 2 bayan arkadaşı ile birlikte <A>
0.86 - <A> hastanesi NEDate bedrettin ulusoy , “ <Q>
0.86 - zel <A> hastanesi NEDate bedrettin ulusoy , “ <Q>
0.86 - <Q>'nun babası NEPersonName , bugün öğle NETime özel <A>
0.86 - <Q>'nun babası NEPersonName , bugün öğle NETime özel <A> hastanesi
0.86 - <Q> (june 29 , 1979 amasra , NECityName - july 4 , 2007 <A>
0.86 - <Q> (june 29 , 1979 amasra , NECityName - july 4 , 2007 <A> ,
0.86 - <Q> (d . 29 NEDate , NECityName ö . NEDate , <A>
0.86 - <Q> (d . 29 NEDate , NECityName ö . NEDate , <A>)
0.86 - <Q> tedavi gördüğü özel <A>
0.86 - <Q> tedavi gördüğü özel <A> hastanesi
0.86 - <Q> , NEPersonName ve NEPersonName inanın , NEDate sabahı <A>
0.86 - <Q> , NEPersonName ve NEPersonName inanın , NEDate sabahı <A>

ulucanlar

PlaceOfDeath - AnswerPatternRaw

0.97 - <Q> bulvarı no : 5 06100 emek / <A>
0.93 - <Q> bulvarı no : 5 kat : 1 / 108 emek / <A>
0.93 - <Q> canım <A>
0.93 - <A>'da geçirdiği trafik kazasında hayatını kaybeden ünlü rock şarkıcısı <Q>
0.93 - , <A>'da geçirdiği trafik kazasında hayatını kaybeden ünlü rock şarkıcısı <Q>
0.92 - <Q> canım <A> video
0.92 - <Q> , hastalanması üzerine tedavi için gönderildiği <A>
0.92 - <Q> sanki hala <A>
0.92 - <Q> sanki hala <A>'
0.92 - <A> hastanesi'ne kaldırılan <Q>
0.92 - zel <A> hastanesi'ne kaldırılan <Q>
0.92 - <Q> - tarlaya ektim soğan - <A>
0.92 - <Q> - tarlaya ektim soğan - <A> konseri
0.92 - <Q> , 27 şubat 1947'de <A>
0.92 - <Q> , 27 şubat 1947'de <A>'
0.91 - <Q> cd . (köroğlu) no : 14 / 3 gop / <A>
0.90 - <Q> belgeseli " galası 17 mart'ta <A>
0.90 - <Q> belgeseli " galası 17 mart'ta <A>'
0.89 - <Q> , hastalanması üzerine tedavi için gönderildiği <A>'
0.89 - <Q> , yanındaki 2 bayan arkadaşı ile birlikte <A>'
0.88 - <Q> bulvarı no : 5 kat : 1 06100 emek / <A>
0.88 - <Q> bulvarı no : 5 kat : 1 / 108 emek / <A> adresinde
0.88 - , <A>lı olan <Q>
0.88 - <A> gop <Q>
0.88 - <A> gop <Q>
0.88 - <Q> 5 mart 1925 yılında <A>
0.88 - <Q> 5 mart 1925 yılında <A>'
0.88 - <Q> 9 eylül'de mülteci olarak yaşadığı <A>

0.88 - <Q> 9 eylül'de mülteci olarak yaşadığı <A>'
0.88 - <A>'daki bir kafede düzenlediği toplantıda , <Q>
0.88 - , <A>'daki bir kafede düzenlediği toplantıda , <Q>
0.88 - <Q> , hüseyin inan ve yusuf aslan , idam edilişlerinin 37 . yılında <A>
0.88 - <Q> , hüseyin inan ve yusuf aslan , idam edilişlerinin 37 . yılında <A> kar
0.88 - <Q> , yusuf aslan ve hüseyin inan idam edilişlerinin 37 . yıldönümünde <A>
0.88 - <Q> , yusuf aslan ve hüseyin inan idam edilişlerinin 37 . yıldönümünde <A> kar

PlaceOfDeath - AnswerPatternStemmed
0.97 - <Q> bulv no : 5 0610 emek / <A>
0.93 - <Q> bulv no : 5 kat : 1 / 108 emek / <A>
0.93 - <Q> canı <A>
0.92 - <Q> canı <A> vide
0.92 - <Q> , hast üzer teda için gönd <A>
0.92 - <Q> sank hala <A>
0.92 - <Q> , yirm seki doğu günü olan 29 hazi 2007 cuma akşa saat 22 : 30 civa , muğl ilin <A>
0.92 - <Q> , yirm seki doğu günü olan 29 hazi 2007 cuma akşa saat 22 : 30 civa , muğl ilin <A> il
0.92 - <Q> - tarl ekti soğa - <A>
0.92 - <Q> - tarl ekti soğa - <A> kons
0.91 - <Q> cd . (köro) no : 14 / 3 gop / <A>
0.90 - <Q> belg " gala 17 mart <A>
0.89 - <A> ilçe geçi traf kaza sonr yara ve teda görd özel hast haya kayb şark ve oyun <Q>
0.89 - l <A> ilçe geçi traf kaza sonr yara ve teda görd özel hast haya kayb şark ve oyun <Q>
0.88 - <Q> bulv no : 5 kat : 1 0610 emek / <A>
0.88 - <Q> bulv no : 5 kat : 1 / 108 emek / <A> adre
0.88 - <A> gop <Q>

0.88 - <A> gop <Q>
0.88 - <Q> 5 mart 1925 yılı <A>
0.88 - <Q> 9 eylü mült olar yaşa <A>
0.88 - <Q> , hüse inan ve yusu asla , idam edil 37 . yılı <A>
0.88 - <Q> , hüse inan ve yusu asla , idam edil 37 . yılı <A> kar
0.88 - <Q> , yusu asla ve hüse inan idam edil 37 . yıld <A>
0.88 - <Q> , yusu asla ve hüse inan idam edil 37 . yıld <A> kar
0.86 - <Q> maha <A> asfa
0.86 - <Q> , 24 ocak 1993 <A> karl
0.86 - <Q> cina önce <A>
0.86 - <Q> tiya sakı saba müze sakm club sals <A>
0.86 - <Q> tiya sakı saba müze sakm club sals <A> sama
0.86 - <Q> (d . 2 ocak 1943 , <A>
0.86 - <Q> (d . 2 ocak 1943 , <A> -
0.86 - <Q> , evli de <A>
0.86 - <Q> , evli de <A> gele
0.86 - <A> ilçe geçi traf kaza haya kayb ünlü rock şark <Q>
0.86 - l <A> ilçe geçi traf kaza haya kayb ünlü rock şark <Q>