Automatic true-false question answering in meetings

Quoc Anh Le

Master Internship Report

University of Namur

Faculty of Computer Science

Advisors: Andrei Popescu-Belis (Idiap Research Institute)

Jean-Paul Leclercq (University of Namur, FUNDP)

August 2008 - January 2009

Abstract

A system for automatic true-false question answering (QA) over meeting transcripts was developed using speaker-directed lexical similarity algorithm including n-grams matching. The main function of this system is to determine the true and false statement in a pair of complementary statements. These statements were created using a defined methodology about a fact related to the meeting for the Browser Evaluation Test method, namely BET questions [1]. Answering BET questions are done by human subjects using a meeting browser in order to evaluate the performance of this browser. Hence, this system is the first step to build an automatic assistant tool that helps humans answer such type of questions.

Most question answering systems use lexical similarity algorithm to locate relevant passages most likely to contain the answer. For this, all passages are compared with each other using passage score that is sum of scores of matched words between question and passage. Word score may be calculated based on its frequency, its part of speech (PoS) or its relation with neighbour words. In our own algorithm, a simple lexical similarity algorithm is developed in which passage score is not only based on matched words but also on speaker of these words to retrieve the most relevant passage. This technique pays more attention to the features of a conversational document as meeting transcript. Based on the retrieved relevant passages, the system gives true-false answers.

The performance of this system is evaluated by answering approximately two hundreds of BET questions, which were constructed by independent observers over two meetings of the AMI Meeting Corpus [2]. Experimental results showed that around 58% of retrieved passages are correct meanwhile the chance of guessing randomly one correct passage is less than 4%. The proportion of correct answers finally achieved is around 61%. This result is better than result of answering true-false questions by chance whose proportion of correct answers is only 50%. In addition, the performance of the algorithm is also evaluated based on ASR transcripts, which were generated by an Automatic Speech Recognition [3], as well as meeting summaries based on ASR transcripts by replacing original transcripts by them. These transcripts are certainly more *noisy*. Thus, the proportion of correct answers reduces reasonably for passage retrieval.

The last evaluation is performed by comparing BET scores by human subjects. These scores are from applying the BET method for Transcript-based Query and

Brower Interface (BET4TQB)[4] with scores obtainned by the system over the same BET questions. Our comparative analysis showed that human subjects generally answer questions that require a deduction better than an automatic question answering system. Furthermore, it should better develop this system as an assistant tool that helps humans answer such type of questions by locating the relevant passage rather than return the final answers.

Keywords: Question Answering, Meeting Browser Evaluation, Passage Retrieval, BET questions, True-False Answering, N-gram Matching, Lexical Similarity.

Acknowledgements

The first person I would like to express my gratitude is my supervisor, Mr. Andrei Popescu-Belis [5], senior researcher at the Idiap Research Institute (Dalle Molle Institute for Perceptual Artificial Intelligence), in Switzerland. In fact, he gradually guided me method in doing research in general and this project in particular. I recognize that I have learned a lot from him. During 6 months of the internship in Idiap, he always encouraged me and this much motivated me and made me enjoy my work.

Secondly, I would like to thank all of professors and teachers at the Faculty of Computer Science, University of Namur (FUNDP) for their interesting and valuable courses, that have equipped me with background knowledge to complete this project. Especially, my special thanks should be sent to Prof. Jean-Paul Leclercq [6], my supervisor at FUNDP . He is not only my teacher, but also a good friend who gives me useful advices for my works.

This project is part of the Interactive Multimodal Information Management project (IM2, http://www.im2.ch), funded by the Augmented Multiparty Interaction Training Programme (AMIDA, http://www.amiproject.org). I wish to send my great thanks again Mr. Andrei Popescu-Belis, Prof. Jean-Paul Leclercq and the Dean of Computer Science Department, University of Namur, Jean-Marie Jacquet [7] for their letter of recommendation so that I was accepted for this project.

In addition, I acknowledge the financial support of the Belgian Development Cooperation Agency (BTC, http://www.btcctb.org) for my 2-year course of Computer Science Master in Namur, Belgium.

Last but not least, I thank my wife and my son for their understanding and support during my studies.

Contents

List of Figures

List of Tables

List of Algorithms

Glossary

- Stem: According to the Summer Institute of Linguistics International http://www.sil.org, a stem or word stem is a root or roots of a word that is common to all its inflected variants.

- Lemma: A lemma in morphology is defined as a canonical form of a set of words that have the same original meaning [9]. A lemma is different from a stem that a lemma of the verb may change when morphologically inflected, however a stem that never changes by a morphology. For example, for the word "modified", its lemma is "modify" while the stem is "modifi" because we have words such as **modifi**cation.

- Morphology: *"The morphology of the language is defined in terms of a set M of relations between word forms"* [10]

- Keyword: A keyword is defined as any important word that will help an automatic machine answer a question [11]

- Part-of-speech tagger: A programme determines category of a word as a noun, a verb, an adjective,... based on definition of the word and the sentence that the word belong to. [12]

- Stopword: It is a word such as "the", "to" or "for" which generally add little or no information regarding the subject matter of a document [13]

- Punctuation marks: According to Todd, Loreto (2000), in a text except letters and numbers, others are punctuation marks.

- Synonym: A synonym of a word is another word that they share at least one sense in common [10].

- Dynamic Search Window: That is a window used for block-matched algorithm in information retrieval by moving this window to all data blocks as possible. A dynamic search window has dynamic parameters (size and step) that depend on input data. [14]. In this case, a window moves to all passages as possible in the transcript to retrieve a relevant passage.

- Passage: A passage can simply be defined as a sequence of words regardless sentences or paragraphs. Some text-based information retrieval systems define a passage as a fixed-length block of words [14].

- Passage Score: In text-based question answering system, the score of a passage is based on score of its words with respect to question words. The score of a question word found in a passage is computed based on definition of this word and/or relations of this word with other words in the text [15].

- Passage Retrieval Algorithm: Its objective is to determine a passage that is the most likely to contain information that help for answering a question [15].

- Meeting Browser: It is a tool that help humans find relevant information from past meetings in multimedia archives of meeting recordings [8].

- BET: Browser Evaluation Test. This is a method for assessing the performance of a meeting browser on meeting recordings [1].

- Observation of interest: They are statements about a fact related to a meeting collected by independent observers in order to perform an evaluation for meeting browsers according to the Browser Evaluation Test (BET) method [1].

- Observers: They watch selected meetings from corpus, to produce a set of *observations of interest*.

- BET Questions: They are questions or *Observations of interest* used in the BET method [8].

- Human Subjects: They are persons who answer BET questions using a meeting browser [1] and their answers are used to evaluate the performance of this browser.

- Corpus: A set of meeting recordings

- Question-Answering System: A text-based system allows users ask a question in natural language and receive an exact and succinct answer in place of a list of documents that may contain the answer [16, 17].

- Deductive Question: It is difficult to answer this type of questions. Firstly, it comes from the difference expression between strings from question and strings from answer. This is the biggest challenges for question answering [18]. Secondly, this question type demands to seek a fact rather a clear explanation in the text, for example for "How" and "Why" questions. They are difficult for all question answering systems [19].

- ASR meeting transcript: Meeting transcripts are generated by Automatic Speech Recognition [1].

- ASR summaries: They are generated by an automated summarizer based on ASR meeting transcripts [20].

- Cross-Validation method: This method is used to test a configuration of a system for an accuracy estimation in the case that the system does not have enough data to test [21].

- N-gram matching: In textual information retrieval, this method is used to estimate similarity between two string by examining all n-grams matchings, where an n-gram is a substring of n words [22].

- TREC: The Text Retrieval Conferences `http://trec.nist.gov/`. This is an series of workshops for a list of different information retrieval research which question answering belongs to.

Chapter 1

Introduction

1.1 Context

Meetings become more and more essential in the work to exchange information and to make decisions. It is necessary to save meeting information such as videos, audios, transcripts, slides in the multimedia archives of meeting recordings so that human users can easily find relevant information from past meetings using a tool, namely meeting browser [8]. A meeting browser is defined as follows: *"A meeting browser is a system that enables a user to navigate around an archive of meetings, efficiently viewing and accessing the full multimodal content, based on automatic annotation, structuring and indexing of those information stream"* [23].

One approach to meeting browsing is to design general-purpose meeting browsers that help human users to locate the information that is searched for [24], for instance, meeting browser named Archivus at the University of Geneva and at the EPFL [25], Ferret in Idiap Research Institute [26], Transcript-based Query and Browsing Interface (TQB) at the University of Geneva [27], etc. However, another possibility is to design browsers that locate information automatically, for instance for verification (fact checking) purposes.

1.2 Goal

The goal of this project is to design such an automatic browser following a question-answering approach, and assess its performance on a set of pairs of true-false statement, which have been initially used to evaluate human-directed browsers.

In other words, the goal is to design and to implement a system that determines the true and the false statement in each pair based on searching facts on meeting transcripts, to evaluate its performance over a set of about two hundreds such pairs (over two recorded meetings), and to compare it with human subjects using existing meeting browsers. A comparative analysis of the system and the human scores on specific questions should indicate whether or not system and humans have the same difficulties answering such questions. This work will thus show whether this system should be developed as a full automatic browser that gives an exact answer for user's question or only help users locate relevant information in meeting recordings as an assistant tool.

The experimental pairs of true-false statements used for this system were created using a defined methodology about a fact related to the meeting for the Browser Evaluation Test method, namely the BET question [1]. The BET method is presented in detail as follows:

The BET method

One method proposed originally by Flynn, M. and Wellner, P. [28] is the Browser Evaluation Test (BET) that evaluates a meeting browser based on user performance rather than subjective judgment. According to the BET, the task of browsing a meeting recording is an attempt to find a maximum number of *observations of interest* in a minimum amount of time [1], in which *observations of interest* is defined as interesting to the meeting participants or to people who missed the meeting. Thus, evaluation of a meeting browser is to collect a set of *observations of interest* and then ask human subjects to verify these observations as binary-choice test questions in a fixed amount of time by using a meeting browser to access the meeting. A *good* meeting browser will help human subjects find a number of correct answers as many as possible and in a duration as short as possible. Thus, information of the answers by human subjects such as answer precision known as *effectiveness* and answer speed known as *efficiency* are used to evaluate the performance of this meeting browser. In detail, an *observation of interest* is formed as a complementary pair of statements, one true and one false about a fact related to a meeting recording and human subjects are asked to determine which statement is true in the pair. The answer precision is calculated by dividing number of correct answers over total answers. In terms of the answer speed, it is computed as the average time to answer a question rather than the average speed. Analyzing these information and comparing them among meeting browsers will give a score for the performance of a meeting browser.

This is a time-consuming method that need investing in collecting and preparing the observations. However, this observation collection is independent to browsers so that the observations can be extended to be used for the evaluation of other meeting browsers in the future. [29].

Stages of the BET method is presented in the Figure 1.1.

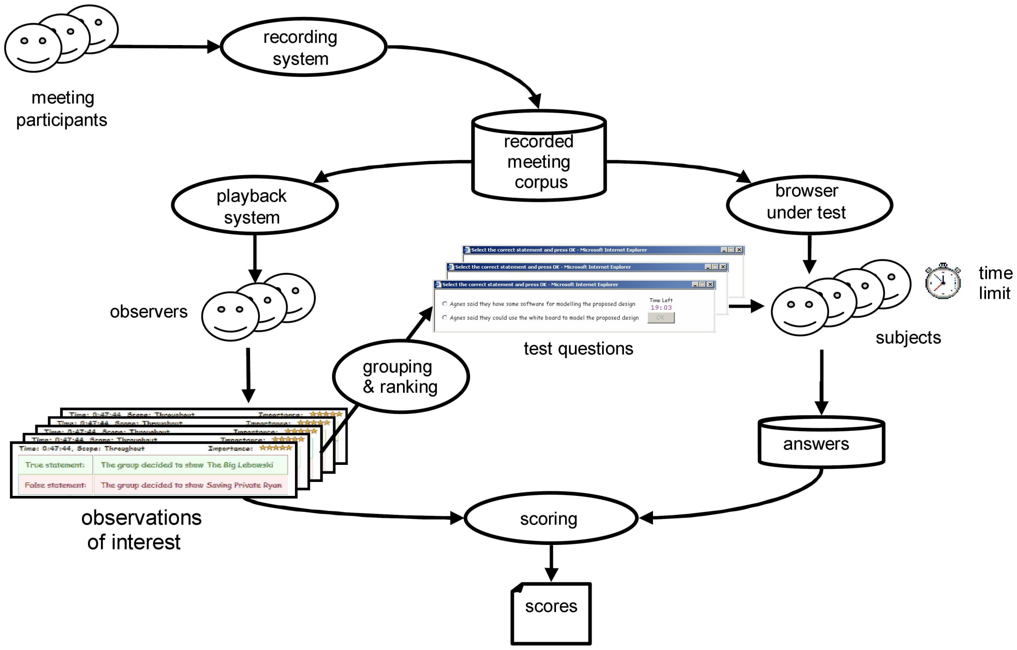

meeting participants
recording system
recorded meeting corpus
playback system
browser under test
observers
Select the correct statement and press OK - Microsoft Internet Explorer
Agnes said they have some software for modelling the proposed design
Agnes said they could use the white board to model the proposed design
Time Left
19:03
OK
test questions
time limit
subjects
grouping & ranking
Time: 0:47:44, Scope: Throughout
Importance: ★★★★★
True statement:
The group decided to show The Big Lebowski
False statement:
The group decided to show Saving Private Ryan
observations of interest
answers
scoring
scores

Figure 1.1: Stages in the design and execution of a BET evaluation [8]

The BET Questions

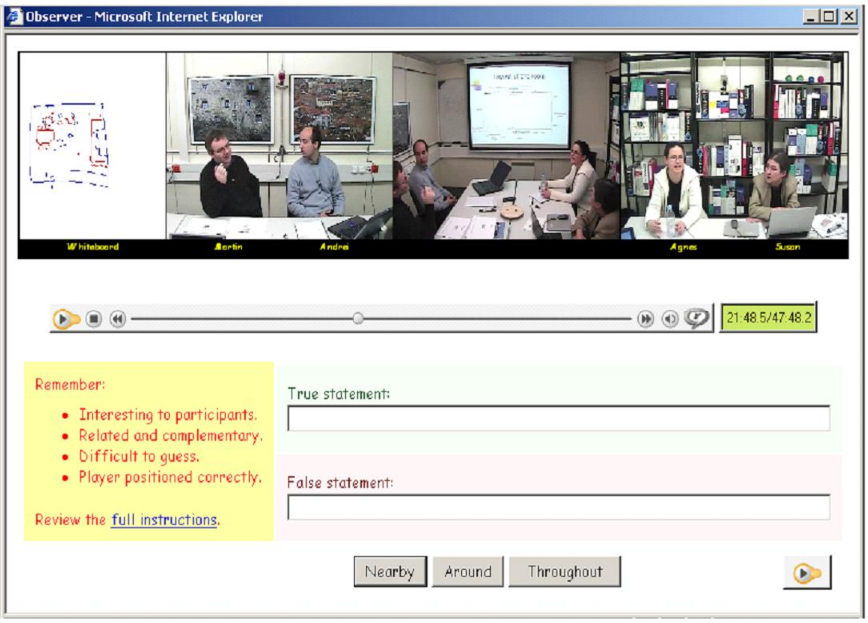
These such pairs of statements to be used for the BET method, called the BET questions, are produced by a set of neutral observers, who independently watch selected meeting from corpus. These observers are native English speakers from the University of Sheffield. They are students, researchers and lecturers. The observers have unlimited time and available full recordings from such media sources as videos, audios, in parallel with paper printouts of the slides that participants worked on for the meeting. At the first time, an observer collects a list of observations as true statements about facts or events that may interest meeting participants or people who missed the meeting. The statements should not be easy to guess without using the meeting information. Then, for each true statement, a false counterpart statement is created so that a pair of complementary statements is generated. The observations should be simple and concisely stated.

An interface for observation collection is presented in the Figure 1.2. As seen in the Figure, there are three buttons ”Nearby”, ”Around” and ”Throughout” that indicate the position of answer information in the transcript. One observation is marked as *Nearby* or *Here* if it is pertinent to that particular moment; marked as *Around* if it covers at least a minute of the meeting around the point the observer have selected; and marked as *Throughout* if it broadly covers the whole meeting. However, in this system, the questions whose type is *Throughout* are avoided because it is difficult to determine relevant passage which contains information of answer for these questions using an automatic system. After that, the collected observations are examined by experts to reject coincided or inappropriate ones.

1.3 Approach

The proposed system has an architecture that is quite common for question answering systems [30] with a number of specificities due to the nature of the data and of the task. The system proceeds in three stages.

The first stage is the pre-processing of the pair of BET questions and of the meeting transcript for the purpose of transforming them into a uniform



Remember:

- Interesting to participants.
- Related and complementary.
- Difficult to guess.
- Player positioned correctly.

Review the full instructions.

True statement:

False statement:

Figure 1.2: Interface for observers

data.

Then the second stage aims at identifying separately the passage of the transcript that is most likely to contain the answer for each question in a pair using lexical similarity algorithm. For this, all passages in the transcript are compared with each other using passage score, which is a sum of scores of matched words between the passage and a question. Regarding the matched word score, it is computed based on a complex score of lexical similarity, in which it is not only based on matched words but also on speaker of these words. This technique pays more attention to the features of a conversational document as meeting transcripts.

At last, the third stage compares two BET statements in the pair based on the paragraph found for each question, and hypothesizes which one is true and which is false.

1.4 Evaluation methods

The performance of this system is evaluated by answering nearly two hundreds of BET questions over two meeting transcripts named IB4010 and IS1008c (see Chapter 3 for more details) from the AMI Meeting Corpus [2]. Furthermore, the performance of the algorithm is also evaluated on ASR transcripts which were generated by an Automatic Speech Recognition [3] as well as meeting summaries based on ASR transcripts. Then the last evaluation is performed by comparing BET scores by human subjects, which are from the BET method for Transcript-based Query and Brower Interface (BET4TQB)[4], with scores obtainned by the system over the same BET questions and the transcripts. This task is to answer the question whether the human subjects and the system have the same difficulties to answer such questions.

Based on these results, the system is estimated to give a conclusion that the system should be developed as a full automated system or an assistant tool that helps humans answer questions concerning meeting information over meeting transcripts.

1.5 Structure

This report contains 7 chapters, in which the first chapter is an introduction while the rest of the report provide information in detail. Accordingly, Chapter 2 previews some available approaches concerning question-answering that are widely applied in many applications. Then, Chapter 3 presents a brief description of two meeting transcripts and the BET questions used to test this system as well as some analysis of data. Chapter 4 consists of three sections that describe three main stages of our approach as mentioned above. In the first section, the pre-processing stage is done in order to normalize the text of transcript and of the BET questions by removing punctuation marks, stop-words, converting abbreviated words into full forms and converting numeric forms into text forms as well as adding some lexical extensions such as synonyms, lemma and stem. The second section corresponds the passage retrieval stage, in which a speaker-directed lexical similarity algorithm including n-grams is presented to locate a passage, which is the most likely to contain information of answer. The way of distinguishing the true statement from another is decided by stage of true-false answer at the last section of this chapter. After that, Chapter 5 mentions an evaluation method using reference answers in order to assess answers returned by the algorithm. Chapter 6 presents our experiments on both manual and automatic (ASR) meeting transcripts. Moreover, at the end of Chapter 6, we conduct a comparison and an evaluation: (i) BET human results are compared with those of the system in order to show limitations of an automatic answering as well as difficulties for both human and machine; (ii) the system and its questions is also used to measure quality of Automatic Speech Recognition (ASR) summaries. Finally, the last chapter gives conclusions.

Chapter 2

Related Work

A question answering system allows users ask a question in natural language and receive an exact and succinct answer in place of a list of documents that contain the answer [16, 17]. Since the first article that addressed a textual question answering system by computer was presented by Simmons (1965)[31], many systems have been developed and some of approaches have been widely used in many applications, for instance Okapi BM2. A typical question answering system is showed in the Figure 2.1.

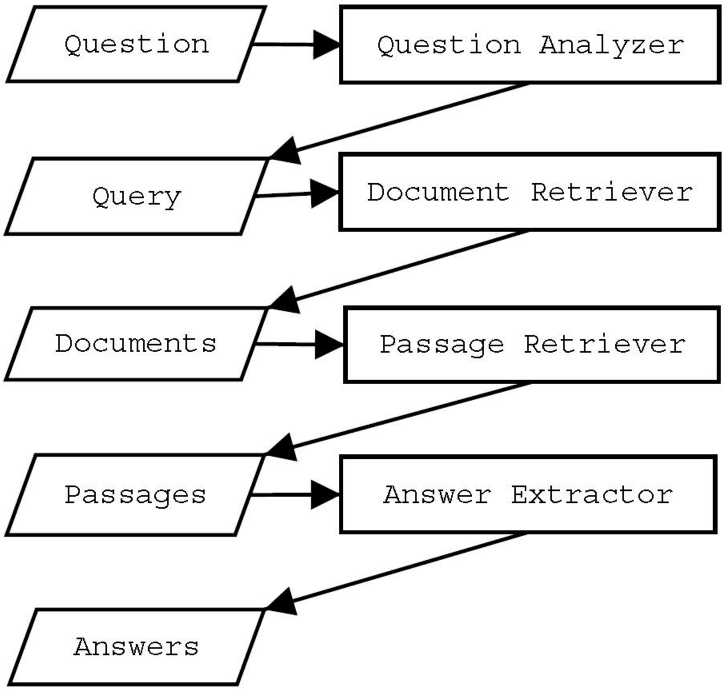


Figure 2.1: A typical Question Answering Architecture

In a general question answering system, there are four major components [15, 16, 32, 33]:

- Question analysis: There are two tasks in this component. Firstly, question in natural language asked by a user need to be converted into queries that are needed by the subsequent parts of the system. The queries created from user's question contain terms likely to appear in documents containning an answer, for instance for such question as *What is the capital of Vietnam?*, the corresponding query is *capital + vietnam*. Secondly, expected type of this question is detected in this stage so that it helps to narrow space of searched answer. For example, questions with "When" always relate to *time*, thus those terms concerning time are remarked such as *date*, *hour*, etc. .

- Document retrieval: This task is to retrieve documents from the corpus that may be taken from Internet by a search engine or archived documents likely to contain answers to the query.

- Passage retrieval: A passage can simply be defined as a sequence of words regardless sentences or paragraphs. Some text-based information retrieval systems define a passage as a fixed-length block of words. [14]. Passage retrieval algorithms take a document and a question, and try to return a list of passages from the document most likely to contain an answer information. The most relevant passages have the highest score, namely passage score, that is generally calculated based on matched words between the document and the question.

- Answer extraction: Based on question analysis and retrieved passages, the system extracts phrase/phrases representing an answer.

We are interested in only passage retrieval stage because in our case, questions and document are defined as the BET questions and meeting transcript as mentioned above. For this reason, we present only state-of-the-art methods related to passage retrieval including both traditional methods and modern methods.

Most passage retrieval algorithms calculate passage score based on words or phases from passage that are found in the question, namely matched words.

However, the way to compute matched word score is different for each algorithm. The simplest algorithm for this approach proposed by Light[30], in which a passage score function counts the number of words from question found in the passage as the score for this passage. That means all words are treated at the same important level. Many questions answering systems use this method as a baseline score to evaluate their performance. Up to date, there are many passage retrieval methods presented in the Text Retrieval Conference (TREC) http://trec.nist.gov. However, these methods can be classified into two groups. One group including traditional methods assigns scores to each matched words independently. That means there are not any relations between two matched words. Another group including modern methods considers relations amongst matched words to assign scores to them. For traditional methods, typical approaches use parts-of-speech and frequencies of a word to give a score for it. Meanwhile for modern ones, dependency relation among matched words in a phrase is computed to give a score for this phrase instead of words. This makes it more semantic than with traditional methods.

Take SiteQ's passage retrieval algorithm [34] as an example, a passage consists of some consecutive sentences segmented by punctuation and passage score is calculated by summing the weights of individual sentence in the passages. Each sentence gets score by a formula that combines both parts-of-speech method and query term density method. In detail, the weight of the matched words is assigned as follows: A proper noun, a common noun recognized by a capital letter has higher score than a verb, an adjective and an adverb. The term density is defined as the distance among matched words. Then if two sentences have the same number of matched words the sentence with smaller distance will have higher score.

The main idea of using word frequencies is that if a word appears many times in a current passage but a few in other passages, its score in this current passage is higher [35]. That means the importance of a word increases proportionally to the number of times this word found in a passage but inversely to the number of times this word found in the total document. The simplest method for this idea is the term frequency inverse document frequency $tf \times idf$. In which, the *tf* is the term frequency that measures the important of the term t_i within the document d_i and *idf* is the inverse document frequency that measures the important of the term t_i in the whole collection of documents. The Okapi BM25 [15, 36–39] presents as state-of-the-art for assigning weights to matched words using word frequencies. In fact, Okapi BM25 is a ranking function that is used to rank matching documents according to their relevance to a given query. Thus, it is rather used in the Document Retrieval stage of the system. However, according to Okapi BM25 presented in the TREC-4 [36], this function is also used for passage determination and searching. In addition, it is a complex function which was developed from the function of term frequency-inverse document frequency *tf-idf* [35].

Modern approaches consider dependency relation among matched words and according to it, n-gram is the simplest case. The n-gram method pays more attention to the order of matching words. Accordingly, those in order is better. More specifically, this method is used to estimate similarity between two strings by examining all n-word substring matching instead of word matching [22]. Another simple method is to use word density presented in the SiteQ's algorithm above. A more complex method for these approaches presented by Cui [40] that uses a dependency tree to assign scores to sentences. Given the reason that one sentence in English can be written in different ways by exchanging position of words in the sentence without chaning its meaning. For instance, with the sentence *Jean wrote a science fiction book*, it may be written in 6 different ways but the meaning remains unchanged. They include: *A science fiction book was written by Jean*, *Jean wrote a book of science fiction*, *A book of science fiction was written by Jean*, *Jean wrote a book of fiction of science* and *A fiction of science book was written by Jean*. These sentences can build a dependency tree that represents correctly position relations of the words in the sentences, so that a given question will be compared with this tree instead of one initial sentence.

In order to enhance the performance of question answering systems, lexical extensions for queries are added such as stemming, lemma, synonyms [30], [32],[33], [34].

Chapter 3

Data description

This system is designed for data with defined format and type so that it has some specifications much clearer compared with those of other question answering systems that normally work with general textual data . In this Chapter, two meeting transcripts and BET questions of these two meetings will be described. An analysis on these data which also give the reasons to build our algorithm is also presented.

3.1 Meeting transcripts

Two meeting transcripts used to test the system are taken from the corpus built by Augmented Multi-party Interactions project [2]. Both of them were discussed in English, involving four participants, native or non-native English speaker. The first meeting IB4010 lasted 50 minutes in which managers of a movie club discussed to select movie for the next shows; meanwhile, in the second one IS1008c, a team discusses the design of a remote control in 26 minutes. There are two versions of tested transcripts, including manual and automatic transcripts.

In addition, some unnecessary information of original transcripts from AMI Corpus are eliminated such as time of utterances, notations of episodes. Consequently, the most important information of two manual meeting transcripts remainned includes speaker name and utterances that are showed in the Table 3.1.

In a conversational document as meeting transcript, information of speaker play an important role to answer questions that verify a statement with respect to the speaker of one/some utterances. That is why in the proposed algorithm, we pay more attention to the name of speakers in both question and transcript, for instance, score for a matched word as speaker name is the highest compared to other matched words.

Table 3.1: Description of meeting transcript

Movie club meeting (IB4010)		**Remote control design meeting (IS1008c)**	
Speaker	**Utterances**	**Speaker**	**Utterances**
Andrei	Hi everyone.	Sridhar	so if you find out from the technology background, okay, so that would be good.
Denis	So I don't know if you all received the the a- agenda for this meeting. Do you - no?	Christine	Sounds good.
Mirek	No, I haven't.	Agnes	Why was the plastic eliminated as a possible material?
Denis	Here it is.	Christine	Because um it gets brittle, .. cracks -
Mirek	Thank you.	Christine	We want - we expect these um uh these remote controls to be around for several hundred years. Good expression.
Agnes	I haven't.	Ed	Good expression.
Denis	So um um the goal for today are um - We have two goals. Uh - First is to decide a movie for uh the next projection for our movie club.	Christine	I don't know, speak for yourself, I'm planning to be around for a while.
Mirek	Mm-hmm.	Agnes	Although I think - $ I think with wood though you'd run into the same types of problems, wouldn't you? I mean, it chips, it- if you drop it, uh it's - I'm not sure $
Andrei	Mm-hmm.	Sridhar	So so you're not convinced* about the the wood, yes.
...	...	...	...

3.2 Questions

The BET Questions have been mentioned above. However, questions used in this system are not full versions of original BET questions which consists of miscellaneous information such as observation time, mediate time, important level, scope, etc. [1]. The system needs only the true and the false statement in each question, which are considered as input data to distinguish one from another. But as a convention, we still use the name "BET question" to indicate these incomplete questions. Examples of some observations are in the Table 3.2.

For two meetings IB4010 and IS1008c, respectively 222 and 217 raw observations were collected by 9 and 6 observers. After being filtered and corrected, the results are only 129 and 158 final pairs of true/false observations correspondingly. However, as mentioned in the Chapter 1, such of questions as *Throughout* are not included in the set of experimental questions. Thus, we have only 116 pairs of true-false statements for IB4010 and 50 pairs for IS1008c.

Table 3.2: Description of the BET Questions

	Movie club meeting (IB4010)	**Remote control design meeting (IS1008c)**
True	Mirek had not received the agenda for the meeting	One of the features under consideration is speech recognition.
False	Andrei had not received the agenda for the meeting	One of the features under consideration is fingerprint identification.
True	None has seen the Shawshank redemption	The product is expected to last over several hundred years.
False	Only two have seen the Shawshank redemption	The product is expected to last more than 5 but less than 15 years.
True	Denis informed the team that the first objective was to choose a film and the second was to discuss an advertising poster	Christine eliminated plastic as too brittle over time.
False	Denis informed the team that the first objective was to choose a film and the second was to discuss a date for the film to be shown	Christine eliminated plastic as it would flex and damage the chips
...	...	...

The questions considered as statements is one main feature that makes this system different to other question answering systems, which normally consists of different questions like "How", "Why", "When", etc.. For this reason, it is not necessary to apply an existing complex algorithm, which is widely used to deal with various type of questions in other question answering systems. Our proposed algorithm is designed therefore to fit type-known questions. In this case, a typical question ask to verify information spoken by a speaker, for instance "Mirek asks who has seen Schindlers List". In this case, they have a speaker name at the beginning of the sentence. According to our statistics, there are over 55% of such questions (28/50 such questions for IS1008c and 87/116 such questions for IB4010). This is an important remark that score of one matched word spoken by speaker whose name is mentioned in both the transcript and the question should be higher than other matched words.

Another feature of the questions is the similarity of two statements in a pair. In most pairs, two statements are different from each other in only one or two words. Therefore, at the Passage Retrieval stage of the proposed algorithm the probability that two corresponding passages for two questions in a pair are coincided is very high. In this case, the true and the false statement can be distinguished by the similarity between each candidate statement with the corresponding passage. In other words, passage score for each statement is compared to determine the true statement/the false statement.

Chapter 4

Proposed algorithm

The system includes in three stages: (i) In the first stage, known as a pre-processing, two questions and meeting transcript are normalized and reorganized in order to enhance the performance of algorithm; (ii) The second stage identifies a section of the meeting transcript which is most likely to contain the answer (i.e. evidence deciding the true and the false statement); and (iii) The third stage compares the two candidate statements with respect to the identified paragraph(s), and returns the true one.

Figure 4.1 gives an overview of the system.

4.1 Pre-processing

There are two main tasks for this section. Firstly, questions and transcript are transformed into the same form of written text so that they can be compared word by word later. Secondly, in order to enhance the probability of matching between two words, each word is extended by lemma, stem and synonyms.

The first processing is done by five operations as follows:

1. Removing characters as punctuation marks: comma, dot, quotation mark, semicolon or exclamation, . . . because these characters do not have any effects on the proposed algorithm based on lexical similarity.

2. Removing stopwords such as ”the”, ”to” or ”for” that generally add little or no information regarding the subject matter of text [13]. This

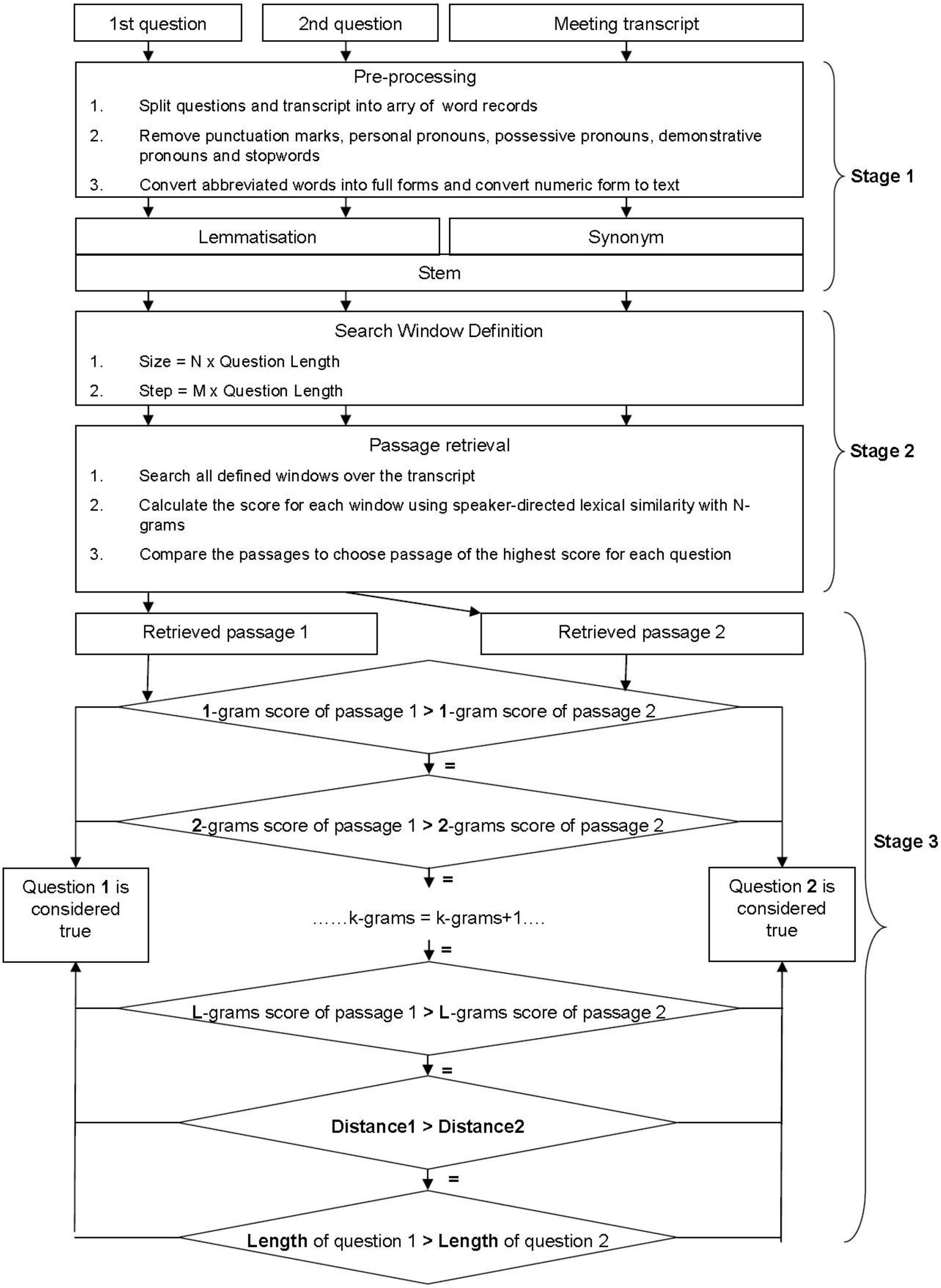

1st question
2nd question
Meeting transcript
Pre-processing
1. Split questions and transcript into arry of word records
2. Remove punctuation marks, personal pronouns, possessive pronouns, demonstrative pronouns and stopwords
3. Convert abbreviated words into full forms and convert numeric form to text
Lemmatisation
Synonym
Stem
Stage 1
Search Window Definition
1. Size = N x Question Length
2. Step = M x Question Length
Passage retrieval
1. Search all defined windows over the transcript
2. Calculate the score for each window using speaker-directed lexical similarity with N-grams
3. Compare the passages to choose passage of the highest score for each question
Stage 2
Retrieved passage 1
Retrieved passage 2
1-gram score of passage 1 > 1-gram score of passage 2
=
2-grams score of passage 1 > 2-grams score of passage 2
=
Question 1 is considered true
……k-grams = k-grams+1….
Question 2 is considered true
=
L-grams score of passage 1 > L-grams score of passage 2
=
Distance1 > Distance2
=
Length of question 1 > Length of question 2
Stage 3

Figure 4.1: Overview of the system

helps to reduce cost of computing as well as to take precaution that they muddle the signal from the more content words [41]

3. For the reason that the questions are indirect-speech statements meanwhile the the meeting transcript is a direct-speech report, words that may be changed for a transformation from direct speech to indirect speech should be avoided from counting matched words between a question and a passage because this leads to lexical mismatches. They are personal pronouns (I, me, myself, you, . . .), possessive pronouns (my, your, . . .), demonstrative pronouns (this, that, . . .).

4. Converting numeric forms to text forms: 34 → thirty four, 2nd → second, etc. . . . so that they are written in the same way, this prevents an unnoticed matching while the algorithm executes.

5. Converting abbreviated words into full forms: We've → We have, I'll → I will, etc. . . . . This operation helps the system treat the text in the same way.

All words are transformed into lower-case forms so that nouns, pronouns, verbs,... have the same important level. That means parts-of-speech will not be used to assign scores to words in the Passage Retrieval stage.

The proposed algorithm uses lexical similarity to find a relevant passage that contains information of answer (Section 4.2). Thus, a lexical extension should be added so that the system have more than one matching possibility between two words. Take the words "production" and "product" as examples, they match each other because they have the same stem.

Three lexical extensions are added to each word, including lemma, stem and synonym.

- Lemma is the canonical form of the word.
- Stem is used to remove inflectional affixes.
- Synonyms have same meaning with the original word.

However, it is not good to applied all these extensions to both questions and transcript. For instance, if both question word and answer word are extended by a set of synonyms, it can create a so call "redundant information".

In addition, it may make the result of the algorithm become wrong if they are considered similar because of similarity of a intermediate synonym. For example: *meeting* and *challenge* have one synonym *contest* in common, but they do not have the same meaning. For that reason, a question word is extended by adding its lemma, meanwhile the transcript words are extended by adding a set of synonyms. Stem is applied as the last operation on each word. This may increase the signal from words, for instance with such a question word as the verb *modified* and a transcript word as the noun *modification*; there are not any matching between these words, even after adding a lemma and a synonym. In detail, *modified* becomes *modify* by a lemmatisation and *modification* has five synonyms "alteration, adjustment, qualifying, limiting, change" returned by WordNet [42]. But after stemming by Snowball [43], these two words will have a same form as *modifi*, so they are matched.

In practice, for each word, the program runs a Stemming API of Porter, called Snowball [43] to obtain a stem and WordNet API [42] to obtain a lemma and a set of synonyms.

A set of synonyms is reduced to have more meaning with the original word by using parts of speech (PoS) tool named QTAG API [12], which aims at removing synonyms that do not have the same PoS with the original word.

Therefore, each word will be treated from now as a record of many fields. For a question word, it has three fields: original word, lemma of the word and stem of the lemma. They are described as the table 4.1. The format of input transcript as described in the Table 3.1 helps the program identify the speaker of any word in the transcript. Hence, a record of transcript word has five fields: original word, stem of word, name of speaker name who spoke this word, set of synonyms and part of speech. They are described in the table 4.2. The name of speakers is also stemmed in order to compare with a question word which may be a speaker name. This field is very important to assess a statement with respect to a specific speaker.

The fields *synonyms* and *lemma* are structured as a set of words because they may have more than one element. For example, lemma of "better" has two words *good* and *well*, synonyms of *better* are *break*, *improve*, *amend*, *ameliorate* and *meliorate*.

For instance, for the pair of such questions as *Mirek had not received the* *agenda for the meeting* and *Andrei had not received the agenda for the meeting*, after removing stop-words, they remain *Mirek had not received agenda meeting* and *Andrei had not received agenda meeting.* Their lexical extensions are presented as following table:

Table 4.1: Word splitting and lexical extensions for questions

Question 1					Question 2			
Position	**Word**	**Lemma**	**Stem**		**Position**	**Word**	**Lemma**	**Stem**
1	Mirek	Mirek	Mirek		1	Andrei	Andrei	Andrei
2	had	have	have		2	had	have	have
3	not	not	not		3	not	not	not
4	received	receive	receiv		4	received	receive	receiv
5	agenda	agenda	agenda		5	agenda	agenda	agenda
6	meeting	meet	meet		6	meeting	meet	meet

One example for a snippet of transcript as below:

```
.....
 Andrei Hi everyone.
 Denis  So I don't know if you all received the the a- agenda for this meeting.
 Denis  Do you - no?
 Mirek  No, I haven't.
....
```

After the processing, it remains:

```
.....
denis   9 not 10 know 11 if 12 you 13 all 14 received 18 agenda 21 meeting
denis   24 no
mirek   25 No 27 have 28 not
....
```

and it is transformed into the following table:

Table 4.2: Word splitting and lexical extensions for transcript

Position	**Word**	**Stem**	**Speaker**	**Synonyms**	**PoS**
...	...				
9	not	not	deni	non	XNOT
10	know	know	deni	cogniz experi live acknowledg recogn ...	VB
11	if	if	deni		CS
13	all	all	deni	entir complet total altogeth whole ...	PDT
14	received	receiv	deni	have get find obtain ...	VBD
18	agenda	agenda	deni	docket schedul agendum ...	NN
21	meeting	meet	deni	fill match ensembl contact ...	NN
24	no	no	deni	nobelium	DT
25	no	no	mirek	nobelium	DT
27	have	have	mirek	receiv get own possess ...	HV
28	not	not	mirek	non	XNOT
...	...				

All steps above are processed in a procedure separated from execution of the main algorithm, which consists of passage retrieval and true-false answer. The questions and the transcript pre-processed from this stage are stored on hard disk so that the programme can load them into RAM before running the main algorithm. This saves us much time to experiment on different configurations of the algorithm that requires the repeatition of execution with the same input data.

4.2 Passage retrieval

The main goal of this stage is to reduce the space of answer being searched by locating a passage which is considered the most likely to contain information about the answer, like evidence that helps to discriminate the true statement from the false statement in a pair. It also gives a numeric score indicating similarity level between found passage and the question so that two complementary statements in a pair may be compared in the subsequent stage.

In order to find relevant passage, the system has to compare all possible passages in the transcript. Information being compared is passage score, which is a numeric value used to measure the similarity between a passage and a question. This task is done by moving a search window from one place to another over the entire transcript. If we assume that the transcript is a cloth stretched to be ironed, then the search window is iron. In the same way of using iron, the search window passes over all possible passages that have the same size with the window in the transcript. At one time, the score of a passage that the window arrives is calculated and then returned in order to compare with the score of passages that window arrived in the past so that the passages of highest score are retrieved [44].

As listed in the Chapter 2, there are many ways to calculate the score of a passage with respect to the question. According to the experimental results of Stefanie [15] with different algorithms of passage retrieval, the performance of the algorithms is different from each others depending on input data. That means each method is suitable for only certain cases. Moreover, existing passage retrieval algorithms use input retrieved by a document retrieval that returns unknown-typed documents such as a forum website, a commercial website or an online newspaper,... Additionally, these approaches have to analyze the type of questions such as ”How” or ”When” before the stage of passage retrieval [15, 16, 45, 46]. This effects final results of passage retrieval algorithm because each step makes an erroneous proportion. In our case, the type of document and the type of questions are defined before, as described in the Chapter 3. That is why we do not apply any existing complex algorithms but develop our own algorithm from the basic one, which was presented by Light et al. [30], by adding some improvements to this algorithm. The method of Light is the simplest method to calculate a passage score that is the number of common words between the passage and the question. Based on this method, we added some certain extensions to the original algorithm, including n-gram matching, lemmatisation, stem, synonyms and various scores assigned to matched word. Lexical extensions such as lemmatisation, stem and synonyms have been presented in the previous stage while the application of n-gram matching and different scores are to be explained specifically in following sections.

Accordingly, the score of a passage is based on the number of matched words as described by Light et al. However, there are three score levels assigned to matched words. As illustrated in the table 4.2, each word from the transcript corresponds to one speaker name. Therefore, if a matched word is spoken by a speaker whose name is addressed in the question, score assigned to this word must be added a bonus.

In our experiment system, as presented in the pseudo code 4.2, different values of score are assigned to a matched word as follows:

1. If a word from the question matches the name of the speaker in the passage, then it receives the highest score (e.g., 4.0)

2. If a word from the question matches a word from the passage (lemmas), and this word is spoken by a speaker mentioned in the question, then it receives the second highest score (e.g.,2.5)

3. Otherwise, if a word from the question matches a word from the passage (lemmas) then it receives the ”normal score” (e.g.,1.0)

4. If a word (lemma) from the question matches one of the synonyms of a word from the passage, then it receives a "low" score (e.g.,0.5)

The numeric values listed above are set by the author based on the intuition about the importance of each matching, and might not be optimal for this task. No automatic optimization (statistical learning) can be attempted as the amount of data was insufficient.

As mentioned above, a search window is used to seek and calculate all possible passages on the transcript. The size of the search window is defined as a multiple of question length. Meanwhile, the distance between two consecutive windows is known as window step and also defined as a multiple of question length. For instance, window size = 5 x question size and window step = 2 x question size. Thus, parameters of search window are dynamic depending on input question length. This method seems suitable to retrieve relevant passage using lexical similarity algorithm because when the length of the question increases, the information that question demands is larger. As a result, it is necessary to enlarge the size of search window.

The passage retrieval algorithm returns a list of passages at the same highest score. Nevertheless, we would like only one passage for the next stage of the algorithm because it is supposed that only one relevant passage corresponds to each BET question. In order to choose the most relevant passage, 2-gram score, 3-gram score,..., n-gram score is applied to reduce the number of passages in the returned list, in which n is the number of question words. N-gram score is calculated as follow: Instead of working with matched words between question and answer, the program will work with matched substrings of n words between them. If two passages have same 1-gram score, their 2-gram score are compared with each other to return higher-score passage. If they have the same 2-gram score, their 3-gram score are compared with each other. Then continuing in the same way until one passage has higher score than others or all n-gram scores were compared with each other. If at the last iteration, they still have same score, the first passage is returned. In fact, n-gram matching is the simplest variant of dependency relation approach among matched words that the matching of words in order is better.

The implementation of calculating passage score is described by pseudo codes 4.2 with comments in detail. There are some remarks for this as follows:

- A passage is defined as a class that has 5 properties: 1) Passage score, which indicates the similarity between this passage with the current question; 2) Position of the passage in the transcript; 3) Size of passage; 4) Set of positions of matched words in the transcript; and 5) Distance among matched words (density). This distance is calculated as sum of absolute distance of any two matched words. This task is done at the end of the passage retrieval to prepare for the True-False Answer stage.

- The priority order is given to ”name matching”, then ”stem matching” and lastly ”synonyms matching” in order to bring the highest score for the current passage.

- The total score of the passage is the sum of the scores for each matched word.

- The frequency of one matched word is not used to increase the score. However, in the case of ”multiple matching”, if one word is repeated more than one time in both question and passage, the number of matched words will be counted as the minimum number of appearance of this word between the question and the passage.

Algorithm 4.1 Passage Retrieval

```
Require: Question //Array of word records as described in the table 4.1
Require: Transcript //Array of word records as described in the table 4.2
Require: WindowSize //Size of search window (in word unit)
Require: WindowStep //Distance between two consecutive windows (word unit)
$Passage \leftarrow Empty$ //Initiate a new passage as current passage(the passage class is defined above)
$BestPassage \leftarrow Empty$ //Initiate a new passage as the best retrieved passage at one moment
$Position \leftarrow 0$ //Initialized position of the search window
while $Position < Transcript.length - WindowSize$ do //Search for all passages to choose the best passage
    $Passage.position \leftarrow Position$ //Position of current passage
    $Passage.size \leftarrow WindowSize$ //Size of current passage
    $Passage.WordList \leftarrow Transcrip[Postion \div Position + WindowSize]$ //List of word records is extracted from the transcript[Position,Position+1,..,Position+Size of window]
    $Ngrams \leftarrow 1$ //Firstly, passage score is calculated using unigram matching
    $Passage.MatchedList \leftarrow getMatchedList(Passage, Question, Ngrams)$ //that is from the procedure 4.2
    $Passage.score \leftarrow getPassageScore(Passage, Question, Ngrams)$ //that is from the algorithm 4.2
    if $BestPassage.score < Passage.score$ then //The current passage is better
        $BestPassage \leftarrow Passage$ //remember it as the best passage
    else if $BestPassage.score \equiv Passage.score$ then //If they have the same score
        while $BestPassage.score \equiv Passage.score \wedge Ngrams \leqslant Question.length$ do //Recalculate their score using bigrams, trigrams,.. until their score is different from each other or Ngrams is over the length of question
            $Ngram \leftarrow Ngrams + 1$ //Increase Ngrams by 1
            $BestPassage.score \leftarrow getPassageScore(BestPassage, Question, Ngrams)$ //Recalculation with new Ngrams
        end while
        if $BestPassage.score < Passage.score$ then //Finally, if current passage have better score, remember the current passage as the best passage until now
            $BestPassage \leftarrow Passage$
        end if
    end if
    $Position \leftarrow Position + WindowStep$ //Move the search window forward
end while
$Passage.distance \leftarrow 0$
for $i = 0$ to $Passage.WordList.Length - 1$ do
    for $j = i + 1$ to $Passage.WordList.Length$ do
        $Passage.distance \leftarrow Passage.distance + abs(Passage.MatchedList[i] - Passage.MatchedList[j])$
    end for
end for
return BestPassage
```

Algorithm 4.2 Passage Score Calculation

```
Require: $Question[] \neq null$ //Array of word records as described in the table 4.1
Require: $Passage.WordList[] \neq null$ //Array of passage word records as described in the table 4.2
Require: $Ngrams \geqslant 1$ //1 for unigram; 2 for bigram; 3 for trigram
Score ← 0 //Initialization value for passage score.
Speaker ← Null //Name of a speaker that is mentioned in the question
PositionsSet ← Null //Set of matched word positions
AvailQues[] ← True //Array of available status for each question record
AvailPas[] ← True //Array of available status for each passage record
//Firstly, search for a speaker name that is containned in both question and passage
for $i = 0$ to $Question.length$ do
    $Matching \leftarrow False$
    $j \leftarrow 0$
    while $!Matching \wedge j \leqslant Passage.WordList.length$ do
        if $Question[i].Stem \equiv Passage.WordList[j].Speaker$ then //If one exists
            $Score \leftarrow Score + 4.0$ //Then passage score increases 4.0 points
            $Speaker \leftarrow Question[i].Stem$ //Remember this name for step later
            $AvailQues[i] \leftarrow False$
            $Matching \leftarrow True$
        end if
        $j \leftarrow j + 1$
    end while
end for
//Checking for a N-grams matching
for $i = 0$ to $Question.length$ do
    $Matching \leftarrow False$
    $j \leftarrow 0$
    while $!Matching \wedge j \leqslant Passage.WordList.length \wedge AvailQues[i] \wedge AvailPas[j]$ do
        $Matching \leftarrow True$
        for $k = 0$ to $Ngrams$ do
            if $Passage[j + k].stem \nsubseteq Question[i + k].lemma$ then
                $Matching \leftarrow False$
            end if
        end for
        if $Matching$ then //If one exists
            if $Speaker \equiv Passage.WordList[j].Speaker$ then //Speaker of matching words is mentioned in the question
                $Score \leftarrow Score + 2.5$ //Then the score increases 2.5 points
            else
                $Score \leftarrow Score + 1.0$ //Other cases, the score increases 1.0 point
            end if
            $AvailQues[i] \leftarrow False$ //These words will be disable from next matching process
            $AvailPas[j] \leftarrow False$
            $PositionsSet \leftarrow PositionsSet \cup j$ //Save position of matching to list
        end if
        $j \leftarrow j + 1$
    end while
end for
//Checking for a synonym matching
for $i = 0$ to $Question.length$ do
    for $j = 0$ to $Passage.WordList.length$ do
        if $Question[i] \subseteq Passage.WordList[j].Sysnonyms \wedge AvailQues[i] \wedge AvailPas[j]$ then
            $Score \leftarrow Score + 0.5$
            $PositionsSet \leftarrow PositionsSet \cup j$ //Save position of matching to list
        end if
    end for
end for
return Score, List
```

4.3 True-False Answer

Based on two retrieved passages corresponding to two input questions from the previous stage, the goal of this stage is to identify the true question. In other words, this task determines which true statement in the pair is.

At the first sight, this system seems simpler than other question answering systems, which must analyze the type of question, such as *Who* or *How* before extracting answer words in retrieved passages as mentioned in the Chapter 2. In our case, two questions in a pair are formed as statements and the answer is simply one bit *true* or *false* for each question. However, because two questions are very close with each other, that means in most cases, they are different from each other by only one or two words, it is not easy to distinguish one question from another, even when the retrieved passage corresponding to the true question is correct (the retrieved passage corresponding to the false question is always false because information of this statement does not exist in the transcript).

For passages retrieved from the previous stage, we have two cases: passage corresponding to the true question is evaluated as incorrect or correct. In the first case, when both retrieved passages are incorrect, thus we can not identify the true statement by any reasonable algorithm. The reason lies in the fact that the database, which are the passages, used to reason out the answer is unconfident. In the second case, we also have two possibilities: (i) two passages are identified; and (ii) two passages are different from each other. For the second possibility, there is not relation between the first question and the second question. They are totally different. Their retrieved corresponding passages are therefore two different ones. Then the only way to answer which question is correct is to measure independently the correctness of each question based on their passage and then the question whose correctness is higher is considered to be true. At this time, how to measure them? If we use passage score to evaluate them, there is not any reason that the number of matched words of the false question is less than those of the true question. So this step needs a deeper analysis on semantic text to answer the question persuasively. We have not yet found an effective method to assess which question is ”more” correct.

Proposed algorithm is applied rather in the case that both the two passages are identified. That means the same passage retrieved for both the true and the false candidates as well as the passage corresponding to the true question are correct. In this case, it is more probable that passage score of the false statement is less than this of the true statement. It is also the main idea of the proposed algorithm for this step. Despite the simplicity of this algorithm, experimental results are much better than results by chance (see section 6.2).

In detail, all scores for 1-gram matching, 2-gram matching, ..., n-gram matching of two passages are used to assess two questions. If two passages have the same 1-gram score, their 2-gram score will be compared, continuing in the same way until one passage have higher score or all n-gram scores were compared. In this case, n is number of words from the smaller question. In fact, using n-gram matching is the simplest version of using dependency relations between neighbor words. The way of calculating n-gram matching has been presented in the previous section Passage Retrieval. In the case they still have the same score, the number of words from two passages are then compared with each other. To explain, one question will be considered to be true if the number of matched words over the total number of question words is smaller.

Pseudo code of the algorithm is described as follows:

In such case, the distance among matched words is calculated in the previous stage, (see the pseudo codes 4.1).

For instance, with two questions presented in the table 4.1, the algorithm gives the following results:

Passage score for the first question *Mirek had not received agenda meeting* is 12.5, corresponding to the retrieved passage below:

```
denis   9 not 10 know 11 if 12 you 13 all 14 received 18 agenda 21 meeting
denis   24 no
mirek   25 No 27 have 28 not
```

In this case, the score for such five matched words as *not*, *receive*, *agenda*, *meet* and *have* is 1.5 while the score for speaker name matching *Mirek* is 4.0. However, because the word *have* is spoken by *Mirek* and the word *Mirek* is found in the question, a bonus 1.0 is added to the total score. Thus, the total score is 12.5.

Algorithm 4.3 True-False Answer

```
Require: $Passage1$ //That is the best passage for question1 which is from the algorithm 4.1
Require: $Passage2$ //That is the best passage for question2 which is from the algorithm 4.1
Require: $Question1$ //Array of word records as described in the table 4.1
Require: $Question2$ //Array of question2 word records as described in the table 4.1
  if $Passage1.score > Passage2.score$ then //If the score of passage1 is higher than that of passage2
    return 1 //The question1 is considered as one true statement
  else if $Passage1.score \equiv Passage2.score$ then //If they have the same score
    if $Passage1.distance < Passage2.distance$ then //compare their distance among matched words as calculated in the algorithm 4.1
      return 1 //The question1 is considered as one true statement
    else if Passage1.distance ≡ Passage2.distance then //If they still have the same distance
      $Ngrams = 2$
      while $Passage1.score \equiv Passage2.score$ do //Recalculate the passage scores using Ngrams matching
        $Passage1.score \leftarrow getPassageScore(Passage1, Question1, Ngrams)$ //That is from the algorithm 4.2
        $Passage2.score \leftarrow getPassageScore(Passage2, Question2, Ngrams)$
        $Ngrams \leftarrow Ngrams + 1$ //Increase Ngrams by 1
      end while
      if $Passage1.score > Passage2.score$ then //If the score of passage1 is higher than the score of passage2
        return 1; //The question1 is considered as one true statement
      else
        return 2; //The question2 is considered as one true statement
      end if
    end if
  end if
```

Meanwhile passage score for the second question is 11.5 corresponding to the following retrieved passage:

```
andrei  4 hi 5 everyone
denis   9 not 10 know 11 if 12 you 13 all 14 received 18 agenda 21 meeting
```

In this case, the score for five matched words *have*, *not*, *receive*, *agenda* and *meet* is 1.5. Meanwhile, the score for speaker name matching ”Andrei” is 4.0. Then, the total score is 11.5.

Consequently, the first question higher passage score is considered as true.

Chapter 5

Evaluation methods

The performance of the proposed algorithm is evaluated based on the results of both principal phases: Passage Retrieval and True-False Answer.

5.1 Passage Retrieval Evaluation

In the first phase, the correctness of a retrieved passage is evaluated by comparing it with a reference corresponding passage, which was annotated by hand before. Information of the reference passage is the position of its first word in the transcript and its size in word. These are corresponding to two properties of a passage defined in the section Passage Retrieval of the previous chapter.

For example for the question and the transcript in the table 4.1 and 4.2, the reference passage found for this question is (25,28) so that it contains only three words "25 No 27 have 28 not". The name of speaker who spoke these words is always integrated as a field of word record, thus it is not necessary to show the speaker name in the reference passage and even in the retrieved passage.

The size of a reference passage is reduced as small as possible, but this reference passage still contains the most essential words to answer the question. For example above, a reference passage may be (9,28) that help us understand the context of answer but the keywords are "25 No 27 have 28 not" spoken by Mirek.

If candidate passage and reference passage have a non-empty intersection, candidate passage is considered to be correct. The number of overlapped words is fixed as one word in this case. If the system is used to help users locate position of answer information, one overlapped word will be accepted as well. For experimental system, the size of retrieved passage is reduced to region which contains matched words instead of size of search window. For above example, with the first question, the size of search window is 5 x number of question words = 35 words. But the size of the found passage is 14 instead of 35 as size of window. The reason is that the position of the passage is defined as the position of the first matched word in the transcript whereas the size is defined as the distance between the first matched word and the last matched word. In this case, the position of the first matched word and of the last matched word are respectively 3 and 17.

5.2 True-false Question Answering Evaluation

It is supposed that we know the true question and the false question in a pair in the input database. Therefore, a question, which is considered to be true by the system, is evaluated by its index and returned by the program. In this case, in the database, the first question in a pair is known as true. Thus, if the system returns 1 as answer, this answer is correct. On the contrary, if the system returns 2 as answer, this answer is incorrect.

5.3 Cross-Validation method

The 5-fold Cross-Validation method [21] is applied to give the average of the best scores for the proposed algorithm. This method is suitable in the case that the algorithm have not enough data to test. Hence, it hides a part of data as unknown data in the future from building a configuration for the algorithm. After that, it uses these hidden data to test the built configuration. The data used to train the system is called training data and the data used to test the system is called test data.

In this case, a configuration is a pair of parameters of search window: size of window and step of window that have been addressed in the section Passage Retrieval. According to the Cross-Validation method, questions are partitioned into 5 subsets, in which four subsets are used as training data and the remainning subset is used as test data. The algorithm is iterated five times so that each subset will be used as test data one time. Thus, final result is the average of results retrieved after running the algorithm five times in this way.

For each iteration, the system returns a pair of parameters of search window that is considered the most suitable for training data. In which, values tested for both search window parameters are from 1 x question size to 13 x question size. A pair of parameters helps the system obtain the highest score on training data, the it will be tested on the test data. The result obtainned on test data is used to estimate the average score of the algorithm.

Chapter 6

Experimental Results

The goal of this chapter is to discuss the experimental results obtained by the proposed algorithm using evaluation methods as presented in the previous Chapter 5, as well as results obtainned from intermediate steps such as Pre-Processing, Parameter Optimisation.

Input data for experiments are two meeting transcript IB4010 and IS1008c with the BET questions for these meetings as described in the Chapter 3. In addition to experimental data, automated transcripts generated by an Automatic Speech Recognition, namely ASR transcripts and some automatic summaries based on ASR transcripts for two meetings are also used.

Because that two meetings may have different difficulties, we will analyse results for each meetings separately.

6.1 Data Processing

Above all, that is the pre-processing stage, questions and transcript are processed by removing unuseful information as punctuation marks, stopwords and pronominal words and being transformed into a standard form to enhance the performance of lexical similarity algorithm as described in the Section 4.1.

After removing the unuseful words, the length of IB4010 is 4872 words compared to 9488 words of the original transcript. Meanwhile, the length of IS1008c is 2059 words compared to 4000 words of the original transcript. It means that nearly half of total of original words are removed. This helps at least the algorithm speed twice when it searches relevant passages in the transcript.

For the questions, after the pre-processing, average of question length is 8 words compared to 12 words of original questions on average. Reducing length of questions play also an important role to speed the algorithm, because it is used as the base unit to define size of the parameters for a search window (see the definition of search window in the section 4.2).

6.2 System Performance

This section shows the results of the algorithm over both IB4010 and IS1008c using evaluation methods presented in the Chapter 5.

Experimental results show performance of both principal phases of the algorithm, Passage Retrieval and True-False Answer. The contribution of each technique in the algorithm such as n-gram matching and speaker-directed scores assigned to matched words are also demonstrated by showing obtainned scores for each technique separately.

In order to insist on effectiveness of the algorithm, the scores obtainned by chance which is also calculated and compared with the results of the algorithm. The "random" scores for true-false answers are 50% because there are only two values "True" or "False" for an answer. The "random" scores for passage retrieval are calculated as bellows: With a defined search window and input data, we can compute the total of possible passages in the transcript based on transcript size, search window size and search window step. The search window moves on the transcript from one place to another. At one time, position of the window defines a passage that have the same position and same size with the window. Thus, the number of window movements is the total number of passages. Two consecutive positions of the search window movement is defined as step of search window. Consequently, the total passages = [(transcript size - window size)/window step] + 1. If a candidate passage and corresponding reference passage are overlapped each other by one word, the candidate passage is considered to be true as agreed and the average of reference passage is 10 words, we have total number of correct passages is 10 x (window size / window step). Therefore, "random" score is calculated by dividing the number of correct passage by the total number of passages. In the section 6.1, we have the average of question size is 8 words, the length of transcript IB4010 is 4872 and the length of transcript IS1008c. From this numbers, calculated "random" score for IB4010 is 1.65% and for IS1008c is 3.9%.

The method of Cross-Validation is used to calculate the average of scores as described in the pseudo codes 6.1.

Algorithm 6.1 Calculate average score based on Cross-Validation method

```
$Score = 0$
for each $TrainingData$ from 1 to 5 do //There are 5 pairs (Training, Test)
  for each $WindowSize$ from 1 x $QuestionSize$ to 13 x $QuestionSize$ do
    for each $WindowStep$ from 1 x $QuestionSize$ to $WindowSize$ do
      Find relevant passage over Training Data by the algorithm 4.1
      Save passage $P\_max$ whose score is highest until now
    end for
  end for
  Using the pair $(size, step)$ corresponding to $P\_max$
  to find relevant passage $P$ over corresponding Test Data by the algorithm 4.1
  $Score = Score + P.Score$
end for
return $\frac{Score}{5}$
```

Results for the two processing stages of the automatic BET question answering algorithm are given in the Table 6.1, for three variants of the algorithm: using only unigram matching when computing the similarity score (and no weighting of speaker-specific words), then with N-gram matching, and finally with additional weighting of matched words spoken by a speaker mentioned in the question, as explained in the Chapter 4.2.

Passage retrieval provides excellent results compared with the chances of randomly locating the correct passage, with scores of 0.55 ± 0.14 for IB4010 and 0.62 ± 0.16 for IS1008c (obtainned with 5-fold cross validation). The automatic system is of course much faster than humans+browsers, at less than 1 s per question.

When combined with the question discrimination, the performance increases only slightly. The expected score should be an average of the score on the questions for which the passage was correctly identified (55% and 62%) with 50% random chance for the questions on which the passage was incorrectly identified, so about 77% for IB4010 and 81% for IS1008c. The fact that the actual scores are lower (though above chance) shows that the algorithm needs improvement for this stage.

Table 6.1: Performance of the algorithm over two meetings

Condition	Passage Retrieval				True-False Answers			
	IB4010		IS1008c		IB4010		IS1008c	
	Acc.	Stdev	Acc.	Stdev	Acc.	Stdev	Acc.	Stdev
Random	0.17	n/a	0.39	n/a	0.50	n/a	0.50	n/a
Unigram matching	0.27	0.15	0.54	0.21	0.37	0.14	0.36	0.21
N-gram matching	0.32	0.15	0.50	0.19	0.43	0.17	0.42	0.11
N-gram + speaker	0.55	0.14	0.62	0.16	0.57	0.06	0.64	0.18

The standard deviation is calculated based on results from Cross-Validation method using formula 6.1 as follows:

$$Standard\ Deviation = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{6.1}$$

In this case, N = 5 corresponding to five test data from Cross-Validation , $\bar{x}$ is the average value of five test data results and x_i is value of each result among five results of test data.

Our proposed algorithm based on the algorithm of Light using n-gram matching and speaker-directed scores assigned to matched words in the passage retrieval demonstrates the effectiveness of these techniques by promised results in the table. Despite of simplicity of the second phase of the algorithm, its final scores are still a lot better than "random" scores.

6.3 Experiments with ASR transcripts

In this section, the performance of the system will be evaluated on ASR transcripts that are generated by an automatic speech recognition. In this case, two ASR transcripts are generated by using the M4 recognition system developed by Krafiat et al [3]. We will evaluate the obtainned results over the ASR transcripts by comparing them with those over the manual transcripts.

According to the report of Karafiat, the quality of these transcripts are not very good. Logically, scores obtainned over the ASR transcripts should be worse than those over the manual transcripts, but remain in a similar range. Experimental results obtainned by our algorithm over the ASR transcripts as described in the table 6.2 show that the results are good as expected. Firstly, the remain word numbers of the ASR transcripts are not much changed compared with those of the manual transcripts. Secondly, although the rate of correct answers over the ASR transcripts reduces logically, but the reduction is only about 8% correct answers compared with the manual transcripts for both principle phases of the algorithm. This may be explained by a fact that when word error rate of the ASR transcripts affects overall text of the transcripts so that the score of all passages reduces together. The algorithm always chooses the passage of highest score for its answer, thus the accuracy is not much changed. That means the algorithm works very well with the ASR transcripts. This is a promising results for building a full automatic assistance tools over ASR meeting transcrips.

For IB4010, passage retrieval accuracy drops to 0.46 ± 0.13 (from 0.55 ± 0.14) and true-false question accuracy drops to 0.52 ± 0.09 (from 0.57 ± 0.06). For IS1008c, passage retrieval drops to 0.60 ± 0.33 (from 0.62 ± 0.16) and true-false question drops to 0.56 ± 0.19 (from 0.64 ± 0.18).

Two following tables present the results in detail. The method of 5-fold Cross-Validation as described in the Section 5.3 is applied to give the average results over two ASR transcripts. The standard deviation are also calculated according to the formula 6.1.

Table 6.2: Experimental results for ASR transcripts

	IB4010 transcript		IS1008c transcript	
	Manual	ASR	Manual	ASR
Original length	9488	9393	4000	3927
Processed length	4872	4624	2059	1957
Passage retrieval	55%±14%	46%±13%	62%±16%	60%±34%
True-false answer	57%±6%	52%±9%	64%±18%	56%±19%

6.4 Results on summarizations

Meeting transcript summary is a transcript that is shorter than the original but it still keeps main information of the meeting.

In this section, the system is tested on summaries of two meeting transcripts IB4010 and IS1008c based on ASR transcripts, named ASR summaries. There are two purposes for this working. Firstly, it aims at measuring the quality of an ASR summary by its scores when it is used to replace the original transcript. Secondly, it also helps to evaluate the proposed algorithm that should give number of correct answers that decrease little by little when the length of summaries reduces because of errors of the summaries.

Moreover, for these purposes, for each ASR summary, we generate a corresponding "random" summary in order to compare scores of ASR summaries with those of random summaries. If algorithm of summarization is good, score schema of its summaries must be different from that of random summaries. A random summary is created by repeating the elimination of a transcript word randomly until this summary has the same length with the ASR summary. In order to increase precision of results over random summaries, for each known summary length, we create 100 random summaries and calculate their average scores.

The ASR summaries used in these experiments were created by an automatic summarization system presented by Gabriel Murray and Steve Renals [20] using term-weights. In which, each dialogue act is ranked by a score as its level of important, namely ranking score. Based on these ranking scores, we create different summaries by eliminating utterances whose ranking score is less than a defined threshold.

In reality, we define 10 different thresholds. Obtainned results are showed in two tables 6.3 and 6.4 for IS1008c and IB4010. In the tables, the first column is the percentage of summary length compared with the length of original transcript. The second column is threshold that ranking score of all summary utterances is higher or equal. When threshold is zero as the first row, the results are presented for original transcript. Four remain columns are percentages of correct passages and true-false answers for ASR summaries and random summaries correspondingly.

According to the experimental results, we have some remarks as follows:

- The number of correct passages for ASR summaries reduces linearly and more quickly than that of random summaries does when the number of utterances removed from the summaries increases. Graph of summary lengths and number of correct passages for ASR summaries have the same bias, they are parallel with each other. This says that eliminated utterances for ASR summaries contain important information. This is contrary to rule of an automatic summarizer that have to remove firstly utterances whose information is the less important. The random summaries are even better than the ASR summaries according to the results of correct passages. For evaluation of the algorithm performance, the algorithm works well that the number of correct passages reduces linearly for both type of summaries.

- For true-false answers, both type of summaries have the same behaviour. The number of true-false answers reduces logically at the first time. After that it does not decrease but it tends to a random result (the probability of a correct true-false answer by chance is 50%). This is explained by a fact that true-false answers are answered by the algorithm based on comparison between two similarities which are obtainned by considering each question and its corresponding passage retrieved from the phase Passage Retrieval of the algorithm. At first, reducing transcript size affects both both true and false question in a pair, the score of corresponding passages reduces accordingly so that the number of correct answers decreases. However, after that when the difference between the two similarities is not enough to distinguish one question from another or in other words, returned answer tends to be random. This tell that results from the phase True-false Answer are not suitable for aiming to measure the quality of a summary.

Consequently, in this case, the way to eliminate dialogue acts in order to generate a ASR summary did not work very well because it eliminated some important utterances that are necessary to answer the BET questions. For the algorithm, its performance for the passage retrieval stage is stable over all summaries so that it can be used to measure the quality of a summary in this way.

Table 6.3: Results for IS1008c summaries

%Original Length	ASR Summaries			Random Summaries	
	rank score	%cpassage	%canswer	%cpassage	%canswer
100	≥ 0.00	68	64	68	64
85	≥ 0.05	62	62	60	60
74	≥ 0.10	56	54	58	60
64	≥ 0.15	52	52	54	58
57	≥ 0.20	50	48	50	56
54	≥ 0.25	46	48	50	56
52	≥ 0.30	38	46	50	54
49	≥ 0.35	36	46	50	56
45	≥ 0.40	28	48	48	52
41	≥ 0.45	24	46	46	52
38	≥ 0.50	24	46	46	50

Original length of ASR transcript for IS1008c is 1957 words

Table 6.4: Results for IB4010 summaries

%Original Length	ASR Summaries			Random Summaries	
	rank score	%cpassage	%canswer	%cpassage	%canswer
100	≥ 0.00	45	62	45	62
85	≥ 0.05	34	51	44	59
74	≥ 0.10	26	55	38	56
65	≥ 0.15	21	52	37	56
58	≥ 0.20	21	54	37	56
54	≥ 0.25	20	52	36	56
51	≥ 0.30	17	53	36	54
47	≥ 0.35	17	53	36	55
44	≥ 0.40	16	52	33	54
40	≥ 0.45	14	50	35	52
36	≥ 0.50	15	51	33	52

Original length of ASR transcript for IB4010 is 4624 words

6.5 Parameter Optimization

The task of the parameter optimisation is to find parameters which are the best fit for each transcript IB4010 and IS1008c. These parameters are search window size and search window step. The goal is to use obtainned parameters for evaluation methods in the remains of this chapter.

For this, we use 5-fold Cross-Validation method as presented in the previous chapter to build a statistic table. This table consists of columns and rows which present values of window step and values of window size correspondingly, for instance, for position (2,5), the step of search window is 2 x input question size and the size is 5 x input question size. In which, each position (row,column) of the table present number of partitions as training data of Cross-Validation method that obtain maximal scores using the value of parameters corresponding to row and column of this position. For instance, the first partition obtains maximal scores at (2,3), (2,5) and the second partition obtain maximal score at (2,4), (2,5) and the others partitions obtain maximal scores at other pairs of parameters, then value of the position (2,5) of the table is 2 corresponding to two partitions. That means when search window size is 2 and search window is 5, there are two partitions over all five partitions obtain maximal score. The maximal scores are the maximal number of true passages retrieved by the algorithm in the first phase. For this experiment, we only use the first phase Passage Retrieval because it is the most essential of the proposed algorithm.

The following tables present results obtainned for IB4010 and IS1008c in detail:

Table 6.5: Parameter Optimization for IB4010

Step Size	1	2	3	4	5	6	7	8	9	10	11	12	13
1													
2													
3													
4													
5													
6													
7													
8	1			1									
9	4	3											
10	4	2	5*										
11	3	2	5										
12	1												
13		1	2								1		

[*] This position is chosen

Table 6.6: Parameter Optimization for IS1008c

Size \ Step	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
1													
2	1												
3	1	1	1										
4	4*	1	1	1									
5			1	1									
6	2	2	3	1	2	1							
7	1		3		2	1							
8								2					
9	3	1	1	1	1	1							
10	2		1	1	1	1				1			
11	1	1				1	1			1			
12	5	3	3		2	4	3			1	1		
13	1	2	3		3		1			1	1		

* This position is chosen

According to the way of table construction above, parameters are considered good if they help as many training data as possible obtain the maximal scores. Therefore, we will choose parameters at a position whose value is the largest in the table as the relevant parameters. As seen in the table, for IB4010, there are two maximal values at (10,3) and (11,3) and for IS1008c, the value of position (12,1) is the largest. These values of parameters help all training data of Cross-Validation method obtain the best scores. However, it is evident that more size of a search window increases, the higher probability that a passage becomes correct is. When the size of search window is as equal as the size of the transcript, then it certainly contains the information of the question. So the returned passage is always true. In this task, we want to find parameters that help programme obtain the maximal number of correct passages but the objective of the passage retrieval is to reduce the search space. For this reason, we should choose the smallest size of search window that is suitable for most partitions. In the table of IS1008c, the position (4,1) as a narrower window which is helpful for discrimination seems acceptable that is suitable for 4 over 5 training data. That means the size of search window is 4 times question size and the step of search window is 1 time question size are the best fit for the BET questions and the transcript IS1008c. In the table of IB4010, the pair of parameters (10,3) has the best value, thus the size of search window is 10 and the step of search window is 3 are chosen as the best fit for the BET question and the transcript IB4010.

As mentioned in the beginning of the section, the chosen parameters are applied in the next sections.

6.6 Comparison with BET scores by human subjects

The main goal of this comparison is to know whether automatic machine and human subjects have the same difficulties to answer the BET questions. By analysing the scores obtainned by the system and BET scores, we can also identify in which case this system is useful to help humans answer the BET questions.

The BET scores used for this comparison are results from BET for the TQB interface [4] that is known as a Transcript-based Query and Browsing Interface which was developed by Andrei Popescu-Belis. The TQB is considered as a meeting browser tool for searching and browsing multi-modal recordings of group meetings. And the BET method is used to evaluate the performance of this meeting browser over two meetings IB4010 and IS1008c.

According to the BET method, human subjects, that did not work with TQB before, were tested by answering the BET questions using TQB. They were 28 students at the University of Geneva, mainly form the School of Translation and Interpreting. Half of the subjects started with IB4010 and continued with IS1008c, and the other half did the reverse order, thus allowing for differentiated results depending on whether a meeting was seen first or second. That means when subjects worked on first meetings, they were trained with the TQB interface, so that they would answer BET questions better on second meeting. In fact, the average of precision is a bit higher for the second meeting. In this experiment, both BET scores as the first meeting and the second meeting are used to compare with results obtainned by the system. However, only 8 BET questions for each meeting are used for this comparison. In which we are interested in only two information of BET scores, they are time average of answering and precision for each answer.

In order to set up configuration of the system, defined parameters of search window are obtainned from previous section Parameter Optimisation. They are (search window size = 10 x question size, search window step = 3 x question size) for IB4010 and (search window size = 4 x question size, search window step = 1 x question size) for IS1008c.

The BET scores by human subjects and scores obtainned by the system are showed in detail in two tables 6.7 and 6.8. In each table, for the scores by human subjects, *Precis1* and *avg time1* in millisecond are average precision and average time as the first meeting, *Precis2* and *avg time2* are average precision and average time in millisecond as the second meeting. For scores obtainned by the system, *#cpassage* and *#canswer* are number of correct passages and number of correct true-false answers correspondingly. However, number of answers for each question is only one. *time* in millisecond is time for answering one question.

Table 6.7: Comparison with BET scores by human subjects for IB4010

curQuid	Humans				System		
	Precis1	avg time1	Precis2	avg time2	#cpassage	#canswer	#time
1	0.93	303.14	0.71	143	0	0	24
2	0.93	105.36	1.00	66.14	1	1	22
3	0.71	118.14	1.00	89.21	1	1	40
4	0.86	207.5	0.86	206.43	1	1	32
5	1.00	64.71	0.93	37	0	1	16
6	0.93	57.79	1.00	53.21	1	1	17
7	0.93	60.93	0.71	52	1	1	24
8	0.71	129.5	0.79	85.29	1	1	19
	0.88	**130.88**	**0.88**	**91.54**	**0.75**	**0.88**	**24.25**

According to the BET for TQB [4], average precision to answer all BET questions for IB4010 is 0.85 ± 0.05 and 0.70 ± 0.10 for IS1008c. Thus, for human subjects, we can divide the BET questions into two groups: easy and less easy. A BET question belongs to easy group as the first group if average precision of its answers as first meeting or second meeting is more than 0.85 for IB4010 and 0.70 for IS1008c, otherwise it belongs to the second group. Meanwhile, for the system, easy group includes all questions that their number of correct passage or number of correct true-false answer is 1. This help us have a standard to compare the BET scores by humans with scores obtainned by the system.

We examine first the results for IB4010. According to the convention above, for human subjects there is only one less easy question that is question numbered 8, meanwhile there are two less easy questions for the system which are questions number 1 and 5. In fact, all of these questions are deductive questions that require rather a comprehension deeply than a search of lexical similarities. In detail, the question numbered 1 is "throughout". That means it is necessary to read all transcript before answering the question. That is why the system could not identify the correct passage using a small search window that does not cover all information of the transcript as it requires for this type of question. Consequently, the true-false answer is determined by chance that is false in this case. The question numbered 5 also require a deduction for its true statement "No one had seen Goodfellas". In the meeting, when all meeting participants said "No" for question "Have you seen Goodfellas?", it is easy for human subjects to understand the answer of participants. However, this is really a difficult task for an automatic system. Therefore, the system identified incorrect passage. Consequently, its true-false answer is determined by chance, that is true in this case. The question numbered 8 is also a deductive question because it is not easy to match the question "I dislike Quentin" with the text "I am not a huge fan of Quentin". However, the system gave true answer for this question meanwhile it made difficult to human subjects. That is because the keyword Quentin appears only one time in the transcript and the system based on this word but did not based on the meaning of the essential phrase to identify correct passage.

Table 6.8: Comparison with BET scores by human subjects for IS1008c

curQuid	Humans				System		
	Precis1	avg time1	Precis2	avg time2	#cpassage	#canswer	#time
1	0.86	410	0.93	127.36	1	1	13
2	0.67	298.58	0.86	129.5	1	1	45
3	0.82	78.09	0.93	67.5	1	1	15
4	0.89	80.22	0.93	103.93	1	1	16
5	0.63	66.38	0.69	63.92	1	0	20
6	0.67	44	0.73	62.18	0	0	10
7	1.00	24	0.82	48	1	0	11
8	0.67	66	0.64	93.55	0	1	11
	0.77	**133.41**	**0.81**	**86.99**	**0.75**	**0.63**	**17.63**

For IS1008c, as defined above for easy and less easy questions, there are two less easy questions for human subjects. They are questions numbered 5 and 8. For the system, it answered wrong three questions that are questions numbered 5, 6,7 and 8. In which, the questions numbered 5, 6 and 8 are deductive questions. For question numbered 5, whose true statement is "Agnes express her opinion that ...", the correct passage should be "Agnes: I think ... ". Two different expressions make difficult to understand for both human subjects and the automatic system. Dealing with the question numbered 6, which is "Agnes notes some reasons to not have a display", Agnes showed a list of reasons in the transcript but there is few matched words between question string and answer string. This is similar with the question numbered 8. However, the true-false answer for question numbered 8 is correct by chance. The question numbered 7 is not difficult so that the system gave correct passage but incorrect true-false answer. That means true-false answers by the system are not as stable as correct passage answering

In conclusion, although both human subjects and the system meet difficulties to answer deductive questions but it seems be more difficult for the automatic system. For IB4010, there are 3 deductive questions and the automatic system wrong answers 2 over 3 questions, meanwhile the human subjects have only difficulties to answer 1 over 3 questions. For IS1008c, there are also 3 deductive questions. The system gave wrong answers for all three questions, meanwhile the human subjects have difficulties to answer two questions. In fact, deductive questions are equivalent to How and Why questions that are difficult for all question answering systems [18, 19]. According to experimental results, the results for passage retrieval are more logical than the results for true-false answers. That means the system should be developed to help humans answer BET-typed questions by identifying relevant passage instead of giving final answers. In other words, it is a useful tool for locating answer, not necessarily finding it.

Chapter 7

Future Research

The results of the passage retrieval propose a promising assistant tool for meeting browsers. This automatic tool integrated into a meeting browser help users locate relevant information in the meeting in a quick time, so that they can save time to reason out the answer to a question.

This is the first design for an automatic meeting browser following to question answering approach. Thus, this system can be developed by adding a *answer extraction* stage after the passage retrieval stage in order to extract a short phrase that express the answer instead of giving an answer *true* or *false*. However, this requires a deeply research on semantic text.

Chapter 8

Conclusion

The performance of the automatic true-false question answering system is quite below than that of humans using existing browsers. However, the scores of passage retrieval stage are a lot better than random scores, in a short time (less than 1 s per question).

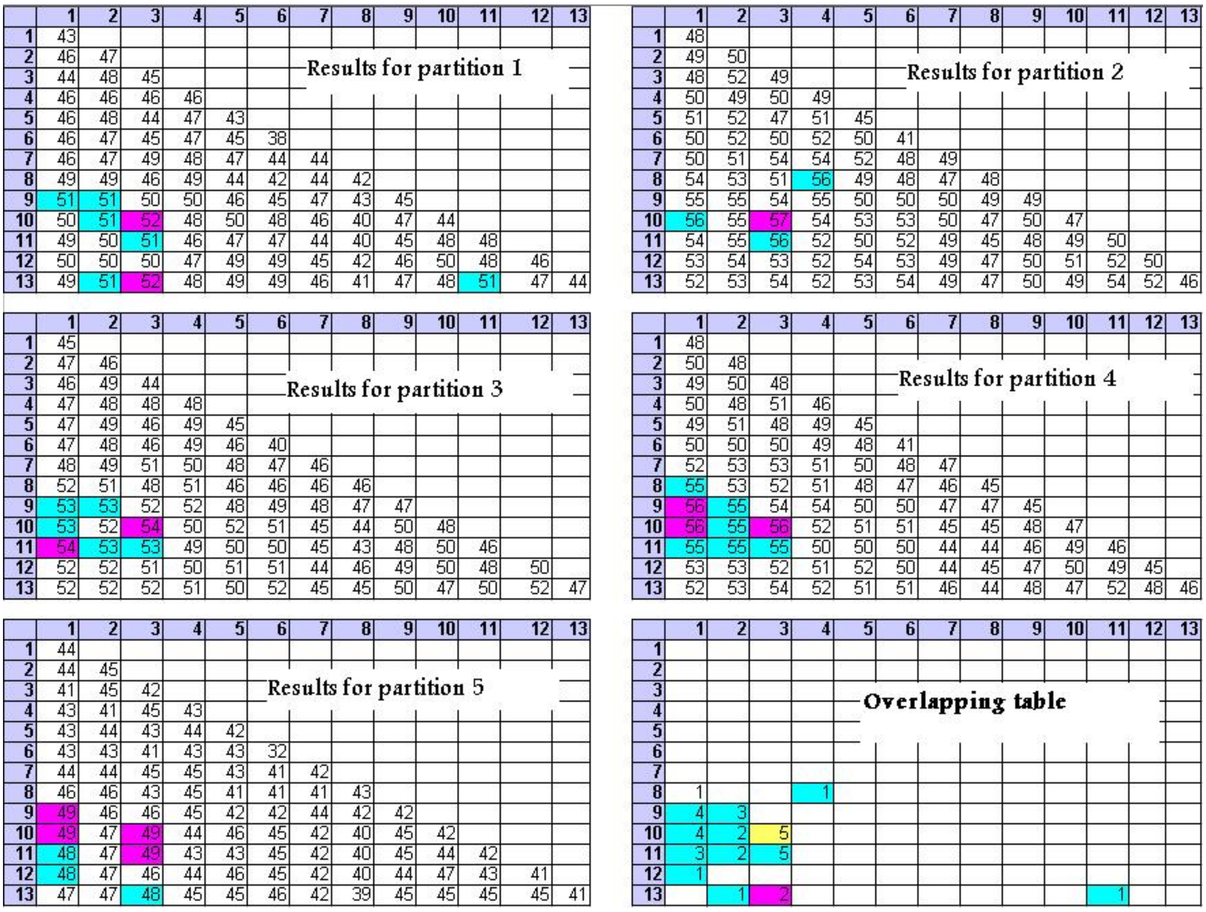
The human subjects answer questions that require a deduction or a reflexion better than the system does but the system gives the answers much more quickly. Thus, the automatic system should give consultative information for a question given by users rather than return the answer in a fully autonomous way.

In conclusion, this project opens a new starting point to study the feasibility of a fully-automatic question-answering system for meeting transcripts. The lexical similary methodology may not be able to solve completely this problem. Nevetheless, it is an open problem and need further researches. Within 6 months of doing this project, I do not have ambition to study and find out a perfect solution but try to experience with simple and basic solutions.

Appendix

8.1 Intermediate results for parameter optimisation

8.1.1 IB4010



Results for partition 1

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	43												
2	46	47											
3	44	48	45										
4	46	46	46	46									
5	46	48	44	47	43								
6	46	47	45	47	45	38							
7	46	47	49	48	47	44	44						
8	49	49	46	49	44	42	44	42					
9	51	51	50	50	46	45	47	43	45				
10	50	51	52	48	50	48	46	40	47	44			
11	49	50	51	46	47	47	44	40	45	48	48		
12	50	50	50	47	49	49	45	42	46	50	48	46	
13	49	51	52	48	49	49	46	41	47	48	51	47	44

Results for partition 2

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	48												
2	49	50											
3	48	52	49										
4	50	49	50	49									
5	51	52	47	51	45								
6	50	52	50	52	50	41							
7	50	51	54	54	52	48	49						
8	54	53	51	56	49	48	47	48					
9	55	55	54	55	50	50	50	49	49				
10	56	55	57	54	53	53	50	47	50	47			
11	54	55	56	52	50	52	49	45	48	49	50		
12	53	54	53	52	54	53	49	47	50	51	52	50	
13	52	53	54	52	53	54	49	47	50	49	54	52	46

Results for partition 3

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	45												
2	47	46											
3	46	49	44										
4	47	48	48	48									
5	47	49	46	49	45								
6	47	48	46	49	46	40							
7	48	49	51	50	48	47	46						
8	52	51	48	51	46	46	46	46					
9	53	53	52	52	48	49	48	47	47				
10	53	52	54	50	52	51	45	44	50	48			
11	54	53	53	49	50	50	45	43	48	50	46		
12	52	52	51	50	51	51	44	46	49	50	48	50	
13	52	52	52	51	50	52	45	45	50	47	50	52	47

Results for partition 4

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	48												
2	50	48											
3	49	50	48										
4	50	48	51	46									
5	49	51	48	49	45								
6	50	50	50	49	48	41							
7	52	53	53	51	50	48	47						
8	55	53	52	51	48	47	46	45					
9	56	55	54	54	50	50	47	47	45				
10	56	55	56	52	51	51	45	45	48	47			
11	55	55	55	50	50	50	44	44	46	49	46		
12	53	53	52	51	52	50	44	45	47	50	49	45	
13	52	53	54	52	51	51	46	44	48	47	52	48	46

Results for partition 5

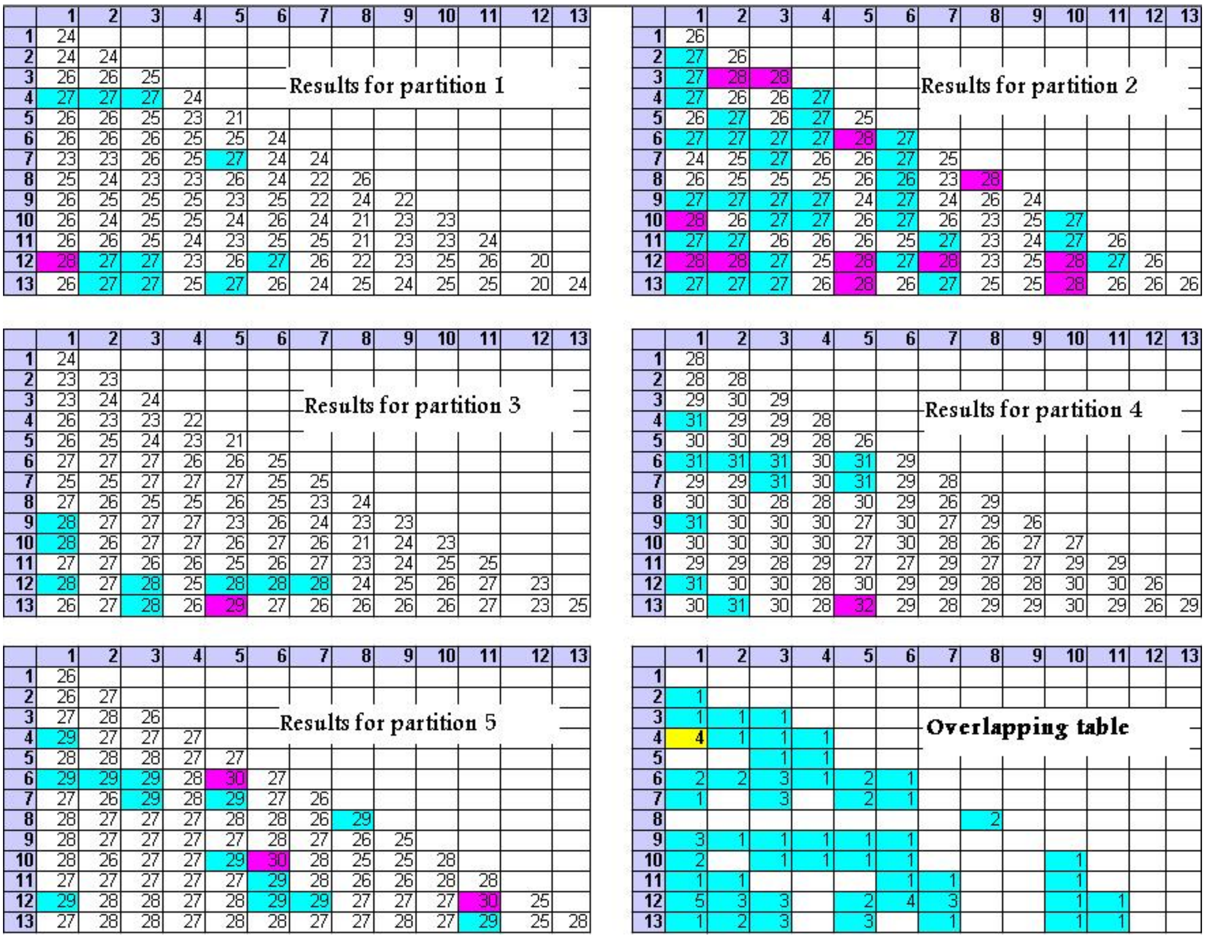
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	44												
2	44	45											
3	41	45	42										
4	43	41	45	43									
5	43	44	43	44	42								
6	43	43	41	43	43	32							
7	44	44	45	45	43	41	42						
8	46	46	43	45	41	41	41	43					
9	49	46	46	45	42	42	44	42	42				
10	49	47	49	44	46	45	42	40	45	42			
11	48	47	49	43	43	45	42	40	45	44	42		
12	48	47	46	44	46	45	42	40	44	47	43	41	
13	47	47	48	45	45	46	42	39	45	45	45	45	41

Overlapping table

	1	2	3	4	5	6	7	8	9	10	11	12	13
1													
2													
3													
4													
5													
6													
7													
8	1			1									
9	4	3											
10	4	2	5										
11	3	2	5										
12	1												
13		1	2								1		

Figure 8.1: Using 5-fold Cross-Validation for parameter optimisation for IB4010

8.1.2 IS1008c



Results for partition 1

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	24												
2	24	24											
3	26	26	25										
4	27	27	27	24									
5	26	26	25	23	21								
6	26	26	26	25	25	24							
7	23	23	26	25	27	24	24						
8	25	24	23	23	26	24	22	26					
9	26	25	25	25	23	25	22	24	22				
10	26	24	25	25	24	26	24	21	23	23			
11	26	26	25	24	23	25	25	21	23	23	24		
12	28	27	27	23	26	27	26	22	23	25	26	20	
13	26	27	27	25	27	26	24	25	24	25	25	20	24

Results for partition 2

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	26												
2	27	26											
3	27	28	28										
4	27	26	26	27									
5	26	27	26	27	25								
6	27	27	27	27	28	27							
7	24	25	27	26	26	27	25						
8	26	25	25	25	26	26	23	28					
9	27	27	27	27	24	27	24	26	24				
10	28	26	27	27	26	27	26	23	25	27			
11	27	27	26	26	26	25	27	23	24	27	26		
12	28	28	27	25	28	27	28	23	25	28	27	26	
13	27	27	27	26	28	26	27	25	25	28	26	26	26

Results for partition 3

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	24												
2	23	23											
3	23	24	24										
4	26	23	23	22									
5	26	25	24	23	21								
6	27	27	27	26	26	25							
7	25	25	27	27	27	25	25						
8	27	26	25	25	26	25	23	24					
9	28	27	27	27	23	26	24	23	23				
10	28	26	27	27	26	27	26	21	24	23			
11	27	27	26	26	25	26	27	23	24	25	25		
12	28	27	28	25	28	28	28	24	25	26	27	23	
13	26	27	28	26	29	27	26	26	26	26	27	23	25

Results for partition 4

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	28												
2	28	28											
3	29	30	29										
4	31	29	29	28									
5	30	30	29	28	26								
6	31	31	31	30	31	29							
7	29	29	31	30	31	29	28						
8	30	30	28	28	30	29	26	29					
9	31	30	30	30	27	30	27	29	26				
10	30	30	30	30	27	30	28	26	27	27			
11	29	29	28	29	27	27	29	27	27	29	29		
12	31	30	30	28	30	29	29	28	28	30	30	26	
13	30	31	30	28	32	29	28	29	29	30	29	26	29

Results for partition 5

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	26												
2	26	27											
3	27	28	26										
4	29	27	27	27									
5	28	28	28	27	27								
6	29	29	29	28	30	27							
7	27	26	29	28	29	27	26						
8	28	27	27	27	28	28	26	29					
9	28	27	27	27	27	28	27	26	25				
10	28	26	27	27	29	30	28	25	25	28			
11	27	27	27	27	27	29	28	26	26	28	28		
12	29	28	28	27	28	29	29	27	27	27	30	25	
13	27	28	28	27	28	28	27	27	28	27	29	25	28

Overlapping table

	1	2	3	4	5	6	7	8	9	10	11	12	13
1													
2	1												
3	1	1	1										
4	4	1	1	1									
5			1	1									
6	2	2	3	1	2	1							
7	1		3		2	1							
8								2					
9	3	1	1	1	1	1							
10	2		1	1	1	1				1			
11	1	1				1	1			1			
12	5	3	3		2	4	3			1	1		
13	1	2	3		3		1			1	1		

Figure 8.2: Using 5-fold Cross-Validation for parameter optimisation for IS1008c

8.2 List of stopwords

i, a, about, above, an, are, as, at, am, and, be, been, being, but, by, do, does, done, did, for, he, her, hers, herself, his, him, himself, how, in, is, it, its, itself, me, my, mine, myself, nor, of, on, or, our, ours, ourself, ourselves, so, she, that, the, they, them, their, theirs, these, themself, themselves, this, those, to, uh, um, up, us, really, very, was, were, we, well, will, with, what, when, where, which, who, whom, whose, why, yet, you, your, yours, yourself, yourselves.

Bibliography

[1] P. Wellner, M. Flynn, S. Tucker, and S. Whittaker. A meeting browser evaluation test. In *Conference on Human Factors in Computing Systems*, pages 2021–2024. ACM New York, NY, USA, 2005.

[2] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, et al. The AMI Meeting Corpus: A Pre-announcement. *LECTURE NOTES IN COMPUTER SCIENCE*, 3869:28, 2006. URL http://www.springerlink.com/index/yg77316944656462.pdf.

[3] V. Wan, M. Karafiát, and S. Renals. Speech Recognition on M4. In *Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, pages 21–23, 2004. URL http://www.dcs.shef.ac.uk/~vinny/docs/pdf/asr.mlmi2004-poster.pdf.

[4] A. Popescu-Belis. Objective Test for Meeting Browsers: the BET4TQB Pilot Experiment. *H., B. and S., R., editors, to appear in Proceedings of Machine Learning for Multimodal Interaction IV, LNCS, Brno, Czech Republic. Springer*, 2007.

[5] Andrei Popescu-Belis. URL http://www.idiap.ch/~apbelis/.

[6] Jean-Paul LECLERCQ. URL http://www.fundp.ac.be/universite/personnes/page_view/01000350/.

[7] Jean-Marie Jacquet. URL http://www.fundp.ac.be/universite/personnes/page_view/01002715/.

[8] A. Popescu-Belis, M. Flynn, P. Wellner, and P. Baudrion. Task-based evaluation of meeting browsers: from task elicitation to user behavior

analysis. URL http://www.lrec-conf.org/proceedings/lrec2008/pdf/580_paper.pdf.

[9] L. Clement, B. Sagot, and B. Lang. Morphology based automatic acquisition of large-coverage lexica. In *Proceedings of LREC04*, pages 1841–1844, 2004. URL http://www.labri.fr/perso/clement/lefff/public/lrec04ClementLangSagot-1.0.pdf.

[10] G.A. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995. URL http://portal.acm.org/citation.cfm?doid=219717.219748.

[11] D. Buscaldi, J.M. Gomez, P. Rosso, and E. Sanchis. N-Gram vs. Keyword-Based Passage Retrieval for Question Answering. *LECTURE NOTES IN COMPUTER SCIENCE*, 4730:377, 2007.

[12] O. Manson. QTAG–A portable probabilistic tagger. *Corpus Research, The University of Birmingham, UK*, 1997.

[13] J. McKechnie, S. Shaaban, and S. Lockley. Computer assisted processing of large unstructured document sets: a case study in the construction industry. In *Proceedings of the 2001 ACM Symposium on Document engineering*, pages 11–17. ACM Press New York, NY, USA, 2001. URL http://portal.acm.org/citation.cfm?id=502190.

[14] N. Goharian and S.S.R. Mengle. On document splitting in passage detection. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 833–834. ACM New York, NY, USA, 2008. URL http://portal.acm.org/citation.cfm?id=1390528.

[15] S. Tellex, B. Katz, J. Lin, A. Fernandes, and G. Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 41–47. ACM New York, NY, USA, 2003. URL http://portal.acm.org/citation.cfm?id=860445.

[16] L. HIRSCHMAN and R. GAIZAUSKAS. Natural language question answering: the view from here. *Natural Language Engineering*, 7(04): 275–300, 2002. URL http://journals.cambridge.org/production/action/cjoGetFulltext?fulltextid=96168.

[17] T. Kato, F.M. Junichi Fukumoto, and N. Kando. Handling information access dialogue through QA technologiesA novel challenge for open-domain question answering. In *Proceedings of the HLT-NAACL 2004 Workshop on Pragmatics of Question Answering*, pages 70–77, 2004. URL http://acl.ldc.upenn.edu/W/w04/W04-2509.pdf.

[18] E. Brill, J. Lin, M. Banko, S. Dumais, and A. Ng. Data-Intensive Question Answering. *NIST SPECIAL PUBLICATION SP*, pages 393–400, 2002. URL http://student.bu.ac.bd/~mumit/Research/NLP-bib/papers/Brill01.pdf.

[19] J. Prager, E. Brown, A. Coden, and D. Radev. Question-answering by predictive annotation. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 184–191. ACM New York, NY, USA, 2000. URL http://portal.acm.org/citation.cfm?id=345574&dl=GUIDE,.

[20] G. Murray and S. Renals. Term-Weighting for Summarization of Multi-party Spoken Dialogues. *LECTURE NOTES IN COMPUTER SCIENCE*, 4892:156, 2008. URL http://www.springerlink.com/index/8x72j8787044757x.pdf.

[21] R. Kohavi. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 14, pages 1137–1145. LAWRENCE ERLBAUM ASSOCIATES LTD, 1995.

[22] A.M. Robertson and P. Willett. Applications of n-grams in textual information systems. *Journal of Documentation*, 54(1):48–67, 1998. URL http://www.ingentaconnect.com/content/mcb/278/1998/00000054/00000001/art00003.

[23] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, et al. The AMI Meeting Corpus. In *Proceedings of Measuring Behavior*, 2005. URL http://www.idiap.ch/~mccowan/publications/mccowan-mb2005.pdf.

[24] D. Lalanne, A. Lisowska, E. Bruno, M. Flynn, M. Georgescul, M. Guillemot, B. Janvier, S. Marchand-Maillet, M. Melichar, N. Moenne-Loccoz, et al. The IM2 Multimodal Meeting Browser Family. *Interactive Multimodal Information Management Tech. Report, Margtiny, Switzerland*, 2005.

[25] A. Lisowska, M. Rajman, and T.H. Bui. ARCHIVUS: A System for Accessing the Content of Recorded Multimodal Meetings. In *In Procedings of the JOINT AMI/PASCAL/IM2/M4 Workshop on Multimodal Interaction and Related Machine Learning Algorithms, Bourlard H. & Bengio S., eds.(2004), LNCS, Springer-Verlag, Berlin*. Springer, 2004.

[26] P. Wellner, M. Flynn, and M. Guillemot. Browsing recorded meetings with Ferret. *Machine Learning for Multimodal Interaction*, pages 12–21, 2004.

[27] A. Popescu-Belis and M. Georgescul. TQB: Accessing Multimodal Data Using a Transcript-based Queryand Browsing Interface.

[28] M. Flynn and P. Wellner. In Search of a Good BET: A proposal for a Browser Evaluation Test. *IDIAP-COM 03*, 11, 2003.

[29] A. Popescu-Belis, P. Baudrion, M. Flynn, and P. Wellner. Towards an Objective Test for Meeting Browsers: The BET4TQB Pilot Experiment. *LECTURE NOTES IN COMPUTER SCIENCE*, 4892:108, 2008.

[30] M. LIGHT, G.S. MANN, E. RILOFF, and E. BRECK. Analyses for elucidating current question answering technology. *Natural Language Engineering*, 7(04):325–342, 2002.

[31] RF Simmons. Answering English questions by computer: a survey. *Communications of the ACM*, 8(1):53–70, 1965. URL http://portal.acm.org/citation.cfm?id=363707.363732.

[32] S. Tellex and B. Katz. *Pauchok: A modular framework for question answering*. Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 2003.

[33] M.W. Bilotti, B. Katz, and J. Lin. What works better for question answering: Stemming or morphological query expansion. In *Proceedings of the Information Retrieval for Question Answering (IR4QA) Workshop at SIGIR 2004*, 2004.

[34] G.G. Lee, S. Lee, H. Jung, BH Cho, C. Lee, BK Kwak, J. Cha, D. Kim, J. An, J. Seo, et al. SiteQ: Engineering high performance QA system using lexico-semantic pattern matching and shallow NLP. *NIST SPECIAL PUBLICATION SP*, pages 442–451, 2002.

[35] S. Robertson. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation*, 60:503–520, 2004.

[36] S.E. Robertson, S. Walker, S. Jones, MM Hancock-Beaulieu, and M. Gatford. Okapi at TREC-4. In *Proceedings of the Fourth Text Retrieval Conference*, pages 73–97, 1996.

[37] M. Beaulieu, M. Gatford, X. Huang, SE Robertson, S. Walker, and P. Williams. Okapi at TREC-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*, 1995.

[38] X. Xue, J. Jeon, and W.B. Croft. Retrieval models for question and answer archives. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 475–482. ACM New York, NY, USA, 2008.

[39] P.R. Comas and J. Turmo. Spoken document retrieval based on approximated sequence alignment. In *11th International Conference on Text, Speech and Dialogue (TSD 2008)*. Springer, 2008.

[40] H. Cui, R. Sun, K. Li, M.Y. Kan, and T.S. Chua. Question answering passage retrieval using dependency relations. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and de-*

velopment in information retrieval, pages 400–407. ACM New York, NY, USA, 2005.

[41] L. Hirschman, M. Light, E. Breck, and J.D. Burger. Deep Read: a reading comprehension system. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 325–332. Association for Computational Linguistics Morristown, NJ, USA, 1999.

[42] M. Pasca and S. Harabagiu. The informative role of WordNet in Open-Domain Question Answering. In *Proceedings of NAACL-01 Workshop on WordNet and Other Lexical Resources*, pages 138–143, 2001.

[43] MF Porter. Snowball Stemmer, 2001.

[44] Le Quoc-Anh and Andrei Popescu-Belis. AutoBET: towards automatic answering of BET questions for meeting browser evaluation. In *IM2 Annual Review Meeting*, 2008.

[45] E.M. Voorhees. The TREC-8 Question Answering Track Report. *NIST SPECIAL PUBLICATION SP*, pages 77–82, 2000. URL http://trec.nist.gov/pubs/trec8/papers/qa_report.pdf.

[46] E.M. Voorhees and D. Harman. Overview of TREC 2001. *NIST Special Publication*, pages 500–250, 2001. URL http://www-nlpir.nist.gov/trec/pubs/trec10/paper.