Answer Re-ranking
with bilingual LDA and social QA forum corpus

Akihiro Katsura

Submitted for the degree of MSc by Research

University of York
Department of Computer Science

March 2014

Abstract

One of the most important tasks for AI is to find valuable information from the Web. In this research, we develop a question answering system for retrieving answers based on a topic model, bilingual latent Dirichlet allocation (Bi-LDA) [1], and knowledge from social question answering (SQA) forum, such as Yahoo! Answers [2, 3, 4, 5, 6, 7, 8, 9 and 10]. Regarding question and answer pairs from a SQA forum as a bilingual corpus, a shared topic over question and answer documents is assigned to each term so that the answer re-ranking system can infer the correlation of terms between questions and answers. A query expansion approach based on the topic model obtains a 9% higher top-150 mean reciprocal rank (MRR@150) and a 16% better geometric mean rank as compared to a simple matching system via Okapi/BM25. In addition, this thesis compares the performance in several experimental settings to clarify the factor of the result.

Contents

List of Tables

List of Figures

Acknowledgements

First of all, I very much appreciate the supervision of Dr. Suresh Manandhar. Without his competent advice, I could not conclude this thesis.

My assessor, Dr. Daniel Kudenko, has also provided very helpful advice, for which I wish to show my appreciation.

I also thank Dr. Matthew Day, Mr. Taku Kawamoto and Mr. Jamie Birch for their help as my proof-readers. I will make my work understood throughout the thesis thanks to them.

There has been an abundance of mental support from my parents, Junko Katsura and Mikio Katsura, and financial support from my grandfather, Reiji Nogaki, both of which I am grateful for.

In addition, I would like to express my appreciation to the cooperative members of the NLP group: Alexandros Komninos, Ali Karami, Baoguo Yang, Brucu Can, Nils Mönning, and Peter Hines; and also my kind office mates, Abdullah Algarni and Hanting Xie; and all the other people in the AI group.

Lastly, I wish to thank my friends, especially the people belonging to the Japanese Society in the University of York, for their special help and friendship.

Declaration

I hereby declare that this thesis was composed entirely based on my own research, except where otherwise noted.

Akihiro Katsura
University of York

1 Introduction

1.1 Preface

Artificial Intelligence (AI) is one of the most important research topics in the twenty-first century. Computers have become essential to our modern and convenience-oriented life. They are utilised in many situations: to collect information, to manage data, to control something remotely, and many more. Whilst some tasks such as numerical processing can be performed extremely quickly and tirelessly by computers, others are more reliant on the interpretation of a human operator. In such cases AI may help to avoid the bottlenecks and costs associated with a human operator.

In recent years, the demand for AI-based question answering (QA) systems is ever growing. A powerful interface for the user to access knowledge is required, because many people suffer difficulty in finding relevant information in a flood of data on the Internet. When we are looking for some information, we usually give keywords to a search engine such as *Google*. However, the system may know neither which keywords should be emphasised among those given nor which sense of a polysemous keyword is required. These may be very import in order to generate an appropriate result. On the other hand, natural language queries can include information needs [11 and 12]. For example, the keywords 'UK' and 'weather' do not contain information needs while the natural language query "why is the weather always bad in the UK?" has the needs. This feature of natural language queries has the potential to solve some of the problems of keyword-based information retrieval. Additionally, the growth of vocal input methods such as *Siri*, provided by Apple Computer Inc., will accelerate opportunities to use natural language queries. It is natural for users to speak to applications as if they were another human and it is easy for the applications to disambiguate each word in their vocal input.

In the field of question answering, one of the unavoidable problems is bridging the lexical gap between questions and answers. Many researchers have worked on the problem with a statistical machine translation (SMT) model in order to 'translate' questions into answers and have achieved successful results [6, 7, 8 and 9]. In this research, a topic model is instead employed to picture the lexical and semantic correlation between questions and

answers. Vulić et al. [1] proposed bilingual latent Dirichlet allocation (Bi-LDA) to infer similar terms over a Dutch-English document-aligned corpus. An example is shown in Table 1.1. It can be expected that this model will work as a substitute for SMT models on a corpus consisting of question and answer documents. To meet the requirement of the corpus size, we incorporated question and answer pairs extracted from a social question answering (SQA) forum, Yahoo! Answers [2, 3, 4, 5, 6, 7, 8, 9 and 10].

The Bi-LDA model is a variant of latent Dirichlet allocation (LDA) models [13, 14, 15 and 16]. These are generative topic models that generate a set of document pairs based on per-document topic distributions and per-topic term distributions. Figure 1.1 shows how a Bi-LDA model generates question and answer document pairs. For each document, topics are sampled based on the topic distribution of the document and for each of the assigned topics, a term is sampled based on the term distribution of the topic. Per-document topic distributions and per-topic term distributions can then be inferred via Bayesian inference from the observed terms. Essentially, Bi-LDA consists of a pair of LDA models with the main difference that per-document topic distributions are shared between each document pair.

Table 1.1. Similar terms inferred via Bi-LDA model over a Dutch-English aligned corpus [1]

(1) **vlucht** (flight)	(2) **reclame** (advertisement)	(3) **munt** (currency)
airlines	advertising	currency
airline	advertisements	currencies
carriers	placement	parities
overbooked	advertisers	fluctuation
easyjet	advertisement	devaluations
frills	stereotyping	euro
flights	billboards	devaluation
booking	adverts	overvalued
booked	advert	peseta
ryanair	advertise	fluctuations

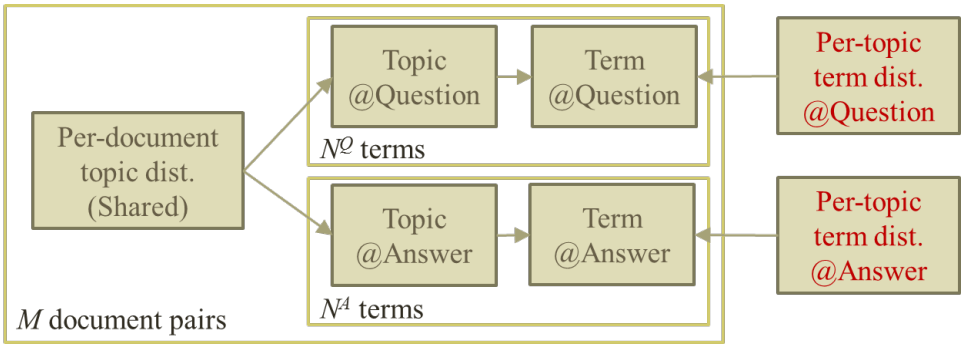

Figure 1.1. A simple diagram of a Bi-LDA model

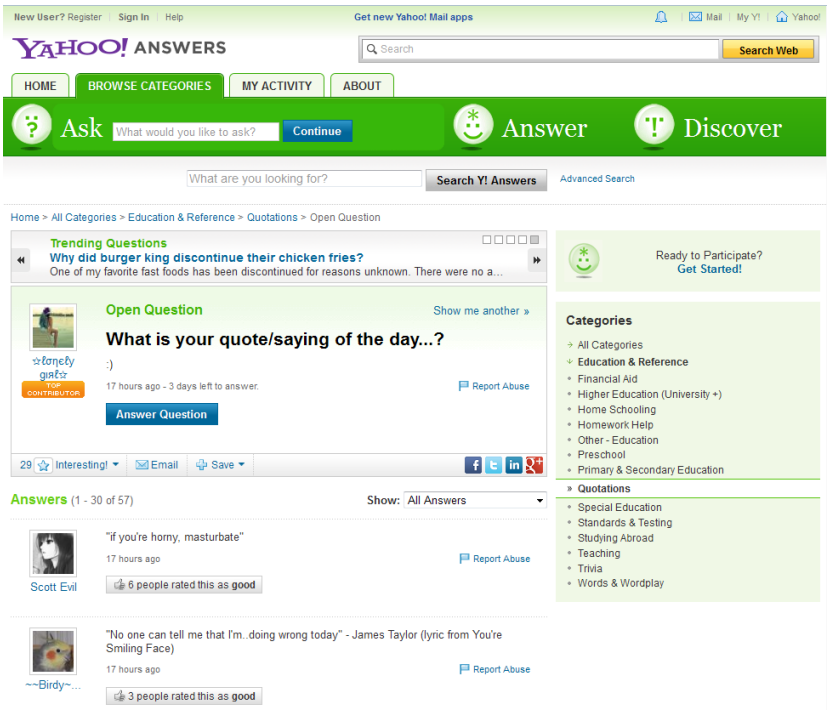

Figure 1.2. A screenshot of a question in Yahoo! Answers

An important feature of Bi-LDA is that the overlap of question and answer per-topic term distributions indicates the vocabulary correlation between questions and answers. Vulić et al. [1] worked on Bi-LDA to rank English words based on the overlap of per-topic word distribution with a Dutch-English document-aligned corpus. Their system successfully found related words from English documents as shown in Table 1.1. In this research, question and answer pairs are used with Bi-LDA inference instead of a Dutch-English aligned corpus.

Internet-based social forums have been previously incorporated in the field of Natural Language Processing (NLP). For this task, SQA forums such as Yahoo! Answers and WikiAnswers are suitable among the social forums because the contents are aligned as question and answer pairs and the number of the instances will be large enough for a Bi-LDA inference. A screenshot of a question and the answers is shown in Figure 1.2. It is not difficult to collect millions of question and answer pairs, for example, from Yahoo! Answers via the APIs provided by Yahoo! Developers. Furthermore, the question and answer instances are open to any domain, and available in many languages. A SQA forum is expected to provide the answers to questions as long as there are reasonable demands for the domain and language of the questions. They also incorporate vote systems, which indicate the quality of the instances. However, instances in a social forum are generally noisy, ill-formed, irrelevant or even incorrect. There are also spam and copy-and-paste posts.

In this research, we incorporate instances extracted from a SQA forum with a Bi-LDA model in order to re-rank answer candidates. The instances are utilised as a bilingual (document-aligned) corpus for the Bi-LDA model. The correlation between question and answer documents is estimated via pointwise mutual information (PMI) metrics and query expansion approaches, by using the knowledge from the topic distributions.

The thesis is organised as follows. In the following three chapters, the background of this research is given. These are the development of QA systems, topic models, and similarity metrics. The fifth chapter shows the mechanism and the performance of the Bi-LDA model to re-rank answer candidate documents. In the sixth chapter, the Bi-LDA model is expanded for taking into account n-grams. Topical terms are then replaced with topic identifiers in order to address the data sparseness problem in the seventh chapter. Finally, the conclusion of this study is given.

1.2 Contributions

In this thesis, a Bi-LDA model with a corpus from a SQA forum was evaluated and we obtained the following results. (1) Applying pointwise mutual information (PMI) to directly score the relevance of documents is not helpful to re-rank answers as compared with expanding queries. (2) The query expansion model with the Bi-LDA model improves

the precision. (3) Three query expansion models are proposed and they outperform an existing model in the experimental settings. (4) N-grams are taken into account for the Bi-LDA model and improve the result of a query expansion approach. (5) Topic-replacement of topical terms into the corresponding topic identifiers improves the PMI based answer re-ranking system whereas it does not improve the query expansion based systems.

2 Question Answering System

To show the basis of the QA systems and their relation to this research, this chapter gives the background of QA systems. The first two sections, 2.1 and 2.2, are focused on the history of QA systems. Section 2.3 clarifies the footing of the system proposed in this research and compares it with existing systems shown in this chapter in order to reveal its originality.

2.1 Early Question Answering Systems

Question Answering (QA) has been a research topic of interest for over half a century. Green et al. developed BASEBALL in 1961 [17], which is an interface to access the database of results of U.S. baseball games by natural language questions. This system firstly analyses the parts of speech and dictionary meanings of words in a question, and infers the type of the question. Then it finds the properties in its database such as date, place and result of game. For example, it interprets a question “Who did the Red Sox lose to on July 5?” as “(To [who]) did [the Red Sox] lose (on [July 5])?” and outputs the loser of the game on July 5 [17].

Another remarkable early QA system is SHRDLU, developed by Winograd in 1972 [18]. This system can be operated in natural language to move blocks in a virtual world. For example, when an operator asks the system “find a block which is taller than the one you are holding and put it into the box”, the system can understand this order, resolving the pronoun ‘it’, and responding correctly. Also, SHRDLU can answer questions about what the system has done, according to the history of its actions and operator’s orders. In the case that the operator asks when the system picked up the block, the system can answer this question and it also can provide the reason for the action on demand. SHRDLU worked very well, but this is because of its virtual world where extremely limited ability is required. The amount of required vocabulary is small, around 50 words, and accordingly almost all sentence patterns can be covered by a careful implementation [19]. Moreover, this small world helps the system to disambiguate questions by evaluating the correspondence with the current state [20 pp. 2].

Early QA systems generally dealt with narrow and specific domains like BASEBALL and SHRDLU, and accordingly they are called closed domain question answering systems. These systems are mainly utilised as natural language interfaces that help users to access a database without knowledge of any data retrieval language (e.g. SQL).

2.2 Current Question Answering Systems

The interest in open domain question answering systems, i.e. those which answer questions of any topic, increasingly grows along with the advance of QA systems and the explosion of the amount of text data on the Internet. However, these systems are really difficult to develop because of the volume of data and the fact that they are not usually structured unlike data in databases used for natural language interface systems.

The Question Answering track of the Text REtrieval Conference (TREC QA track) largely contributed to the development of open domain QA systems. TREC is a workshop organised by the National Institute of Standards and Technology (NIST) in the United States and the QA track was added to TREC in 1999. The track had taken place annually before it was merged into Text Analysis Conference (TAC) in 2008. At first, the task of the TREC QA track was to answer factoid questions. As the workshop evolved, tasks became more difficult, and participants were required to develop systems to answer list, definition and non-factoid questions.

2.2.1 Factoid Questions

Factoid questions are one of the simplest forms of questions which QA systems answer and they are based on a fact, such as “who is the President of the United States?” or “when did Lehman Brothers bankrupt?”. Many QA researchers follow a similar skeleton [21]:

1. Question Analysis
2. Text Retrieval
3. Answer Candidate Extraction
4. Answer Selection or Re-ranking

2.2.1.1 Question Analysis

In the first step, a QA system classifies a user's question into a question type (or perhaps several question types) and also prepares queries in order to retrieve information from the data. Question type indicates what kind of answer a questioner requires. For instance, the question "how high is Mt. Fuji in meters?" requires a number as the answer and similarly "who invented the first aircraft?" is asked so as to find a person's name.

This process may be more difficult than one might first expect. In the early QA systems, regular expressions were used to classify the question type. However, this classification method is too fragile because it is impossible to predict all the patterns of sentences of a user's question considering many factors such as word order, insertion, and synonyms. Also, the interrogative word 'who' indicates not always a person but an organisation. While this task is difficult, it may be the most important task because a failure here affects the remaining three steps.

Question classification is often supported by lexical databases such as WordNet, which show word types, synonyms and parts of speech. Although a syntax analyser is also often used in order to analyse the parts of the speech of questions, there is the problem of computational cost. Then, Finite State Automaton (FSA), or Finite State Transducer (FST), is a possible way to classify question types quickly. The automaton is a finite state machine, in which a state transfers to one of the next states according to the word type in a question like 'who' or 'where', and 'company' or 'country'. According to a state diagram, the automaton determines the question types. An important feature of this model is that the state diagram is human friendly whereas information extracted by Machine Learning approaches is often too abstract for humans to understand. An example of the system utilising a finite state model is FASTUS, developed by Appelt in 1993 [22].

In recent years, the Support Vector Machine (SVM), proposed by Vladimir Vapnik and his colleagues in 1963, has become a widely spread method among QA researchers. SVM tries to linearly classify supervised sample data into positive and negative classes by making a margin as large as possible in the learning phase. This method classifies samples efficiently and is applicable to non-linear boundaries when using kernel functions.

To utilise SVM, QA researchers have to consider how to define feature vectors. N-gram is a reasonable means to produce feature vectors. For example, when processing a question

"which transportation is the most convenient in London?", a bigram feature vector can be defined as (which, transportation, most, convenient, London, which-transportation, transportation-is, the-most, most-convenient, convenient-in, in-London), where 'is', 'the', 'in' and 'is-the' are excluded as stop words. Adding punctuation marks may be helpful to separate a document into chunks and may improve the feature vector. Also, one can include any combinations among the words themselves, parts of speech, word meanings and phrasal expressions [23].

In response to the result of the question analysis, queries for answer retrieval are produced. The format of the queries depends on the specifications of the text retrieval engine employed by the QA system. In the case that the system makes use of a commercial search engine, queries may be simply a series of words with *AND*/*OR* keywords.

2.2.1.2 Text Retrieval

In this step, the QA system retrieves from the data set the documents which are likely to contain the answer to a user's question. A commercial search engine, such as *Google*, or an open source engine, such as *Lucene* and *Indri*, is sometimes used for this step. In particular, *Indri*, developed in the Lemur Project by MIT and CMU, has useful functions for information retrieval, like operators for synonyms and dates when the articles were written. However, there is still a need to develop a system of Information Retrieval for Question Answering (IR4QA) in response to the demands of QA researchers.

Although the means of finding target words is important in this step, this thesis only focuses on how to rank the documents including the words because the ranking greatly contributes to improving the recall of the system.

A typical metric to score candidate documents is TF-IDF. TF stands for Term Frequency, which means the number of the appearances of a term t all over the documents. IDF, Inverse Document Frequency, is the logarithm of the inverse proportion of the documents including the term t. Thus, $\mathrm{TF}(t, \boldsymbol{d})$ and $\mathrm{IDF}(t)$ is defined as

$$\mathrm{TF}(t, \boldsymbol{d}) = \frac{|\{t|t \in \boldsymbol{d}\}|}{\sum_{\forall \boldsymbol{d'} \in \boldsymbol{D}} |\{t|t \in \boldsymbol{d'}\}|}, \quad \mathrm{IDF}(t) = \log_2 \frac{|\boldsymbol{D}|}{|\{\boldsymbol{d}|\boldsymbol{d} \ni t\}|}, \quad \text{where } \boldsymbol{d} \in \boldsymbol{D}.$$

where $\boldsymbol{d}$ is a document and $\boldsymbol{D}$ is the set of documents. The score $\text{TF-IDF}(t, \boldsymbol{d})$ is the product of $\mathrm{TF}(t, \boldsymbol{d})$ and $\mathrm{IDF}(t)$ [24]. There are variants of the normalization factor of TF

but here the frequency of terms in the whole document set is used. $\mathrm{TF}(t, \boldsymbol{d})$ regards the documents which include the term t frequently as important and conversely $\mathrm{IDF}(t)$ penalises the $\mathrm{TF}(t, \boldsymbol{d})$ score if the term t routinely appears in the set of documents $\boldsymbol{D}$. To summarise, the TF-IDF score treats specific documents as important where many of the target terms appear.

Although the IDF modifier looks like a heuristic factor, there is a mathematical background, as formulated in [25]. Let i and j be the indices of a document and a query, respectively. When there are N documents, assuming that each of the N documents will be chosen equally likely as the result of the retrieval, the probability becomes $P(\boldsymbol{d}_i) = 1/N$. Then, the entropy of the candidate documents $\boldsymbol{D} = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \dots, \boldsymbol{d}_N\}$ is:

$$H(\boldsymbol{D}) = \sum_{\boldsymbol{d}_i \in \boldsymbol{D}} P(\boldsymbol{d}_i) \cdot \log\big(P(\boldsymbol{d}_i)\big) = -N \frac{1}{N} \log \frac{1}{N} = -\log \frac{1}{N}$$

On the other hand, for a given term t_j, letting N_j be the number of appearances of the term t_j in the document set $\boldsymbol{D}$**,** and assuming the result document will be selected equally likely, the probability is $P\big(\boldsymbol{d}_i | t_j\big) = 1/N_j$. Given t_j, the entropy of the documents becomes:

$$H\big(\boldsymbol{D} | t_j\big) = \sum_{\boldsymbol{d}_i \in \boldsymbol{D}} P\big(\boldsymbol{d}_i | t_j\big) \cdot \log\Big(P\big(\boldsymbol{d}_i | t_j\big)\Big) = -N_j \frac{1}{N_j} \log \frac{1}{N_j} = -\log \frac{1}{N_j}$$

Thus, the mutual information between the document set and the term set in the query $\boldsymbol{Q} = \{t_1, t_2, \dots, t_M\}$ is:

$$\begin{aligned} I(\boldsymbol{D}; \boldsymbol{Q}) = H(\boldsymbol{D}) - H(\boldsymbol{D}|\boldsymbol{Q}) &= \sum_{t_j \in \boldsymbol{Q}} P\big(t_j\big)\Big(H(\boldsymbol{D}) - H\big(\boldsymbol{D}|t_j\big)\Big) \\ &= \sum_{t_j \in \boldsymbol{Q}} \frac{f_{t_j}}{F}\left(-\log \frac{1}{N} + \log \frac{1}{N_j}\right) = \sum_{t_j \in \boldsymbol{Q}} \frac{f_{t_j}}{F} \log \frac{N}{N_j} \\ &= \sum_{t_j \in \boldsymbol{Q}} \sum_{\boldsymbol{d}_i \in \boldsymbol{Q}} \frac{f_{ij}}{F} \log \frac{N}{N_j}, \end{aligned}$$

where F is the number of all the terms in the document set, f_{t_j} is the number of the terms t_j in the document set and f_{ij} is the number of the terms t_j appearing in the i-th document $\boldsymbol{d}_i$. As this is considerably similar to the formula of TF-IDF, a higher score on TF-IDF

corresponds to a higher mutual information score between a query and a candidate document.

One of a number of variations of TF-IDF is Okapi/BM25, developed by Jones et al. in 2000 [26]. This is commonly used in Information Retrieval research due to its simplicity and accuracy. For Okapi/BM25, TF and IDF are redefined as follows:

$$\mathrm{TF}(t, \boldsymbol{d}) = \frac{(k_1 + 1)|\{t|t \in \boldsymbol{d}\}|}{|\{t|t \in \boldsymbol{d}\}| + k_1 \left\{(1 - b) + b\frac{|\boldsymbol{d}|}{l}\right\}}, \quad \mathrm{IDF}(t) = \log_2 \frac{|\boldsymbol{D}| - \mathrm{DF}(t) + 0.5}{\mathrm{DF}(t) + 0.5},$$

where b is the weight for the importance of the document length, k_1 is the weight for the importance of the frequency and l is the average length of the documents in $\boldsymbol{D}$. Some research such as Isozaki's work [27] has revealed that the term frequency in a document is not important for a factoid QA system because even focal words are often replaced by pronouns and abbreviations.

There are many other measures for efficient and effective text retrieval. For example, bridging lexical chasm [28 pp. 193], i.e. rephrasing synonyms and relevant terms, is a reasonable one because a great number of words have multiple meanings. Also, by making use of the question type, QA systems can avoid irrelevant documents which do not include any answers matching the question type [29].

2.2.1.3 Answer Candidate Extraction

The third step is to extract answer candidates from retrieved documents. Since the type of the user's question is already known, extracting named entities and matching them to the question type is effective in finding the candidates. In the Message Understanding Conference (MUC), which contributed to highlighting the method, named entity extraction, the following seven kinds of named entities are defined [30]:

1. ORGANIZATION
2. PERSON
3. LOCATION
4. DATE
5. TIME
6. MONEY
7. PERCENT

The Information Retrieval and Extraction Exercise (IREX) introduced an ARTIFACT class, in addition to the MUC classification [31]. This class helps, for example, a phrase "Roman Holiday" not to be classified into ORGANIZATION, PERSON, or LOCATION.

If a required set of named entities concerned country names, one could obtain them easily by making use of a country name list. Also, if the entity were a length, one could utilize a regular expression. However, it is generally hard to define rules which manually classify named entities. Thus, machine learning approaches are commonly used for this purpose.

A simple method is a sequential decision process, which determines the named entity classes sequentially from the first to the last term in each sentence. Here, methods must try to avoid contradictions between the term being processed and the previous ones. The Viterbi algorithm [32] is also incorporated in automatic classification of named entities. This algorithm aims to maximise the whole score of classification. It lists possible paths and selects the most likely one with a SVM classifier. These algorithms will fail to classify named entities over a whole sentence if the system makes a single mistake because they do not have a mechanism to re-classify the entities which may be classified incorrectly. An algorithm based on Conditional Random Fields (CRF), proposed by Lafferty et al. in 2001 [33], determines named entity labels probabilistically by considering how the other words are classified.

The fact that named entities often appear as typical patterns helps QA researchers to obtain an answer. Ravichandran and Hovy suggested a way for an automatic acquisition of patterns in order to extract named entities [12]. For instance, answers to the question "When was Gandhi born?" may appear in the following format:

- … Gandhi was born in 1869. …
- … Gandhi (1869-1948) …

Then, if a named entity extraction system has already known that Gandhi was born in 1869, the system can assume that these patterns are useful to find someone's date of birth. However, the system should test these patterns in order to make sure that they are suitable for use in extraction because patterns which can be obtained by this method tend to be too general.

Other useful patterns are the features of apposition and subordination. Pantel and Ravichandran proposed using patterns like “X such as Y” and “X is a Y” so as to collect definitions of words [34]. Cui et al. made use of a soft pattern matching approach to extract definition descriptions in order to increase the recall in TREC-13 [35].

For an accurate QA system, analysis of dependency trees of sentences is important. The kernel tree provides a method for this analysis [10 and 36]. It is easy to perform machine learning via SVM because the tree is defined as a kernel function. Also, researchers can easily calculate similarity between two kernel trees. To measure the similarity [36], at first a tree is partitioned into subtrees with the restriction that partial trees cannot be included. For instance, a noun phrase (NP) must not have a determiner without a noun. Then, the count of common nodes in the subtrees of comparing trees indicates the similarity of the trees. The example of a tree and a set of subtrees are shown in Figure 2.1. The similarity must be normalised since the count is strongly affected by the size of the sentence. In order to employ a dependency tree, a part-of-speech tagger and a dependency parser are required. Both of them are, for example, provided by the NLP group at Stanford University [37 and 38]. Wang, Ming and Chua proposed a “Syntactic Tree Matching” approach, which takes into account the importance of each subtree in order to accurately compute similarities [10].

Anaphora Resolution is another important analysis. Resolving pronouns, such as ‘he’ and ‘it’, and even ellipses plays a significant role for QA systems to interpret user’s questions correctly. In the TREC-13 QA track, participants were required to resolve anaphoric references and ellipses when the TREC organiser added a task to deal with a series of questions. These days, researchers address the problem with machine learning approaches [39].

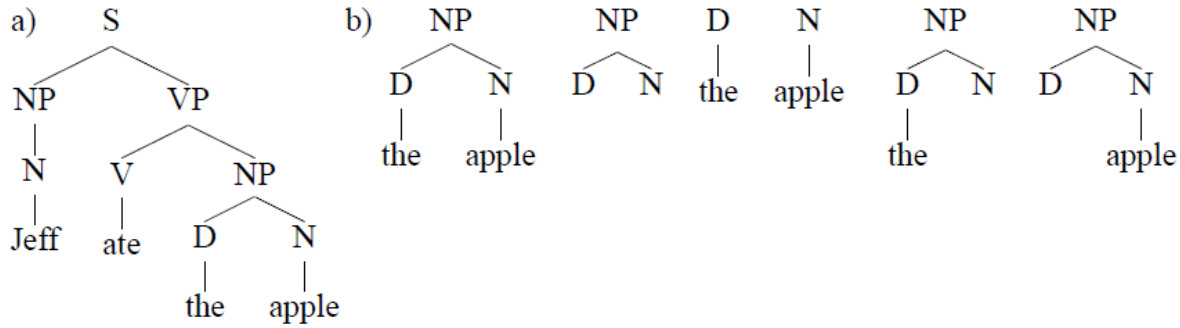

Figure 2.1 (a) An example tree, (b) a set of subtrees [36]

2.2.1.4 Answer Selection or Re-ranking

The correct answer of a question is found with the queries obtained in the step of question analysis, based on the component terms of the question. Heuristics are demonstrated over the TREC QA corpus by Clarke and Terra [40]. Indeed, many QA researchers have utilised distances between the queries and candidate answers to measure the likelihood that each answer is correct. However, the queries are often collocated accidentally and therefore a QA system has to remove irrelevant candidates.

In order to avoid the extraction of irrelevant candidates, it is reasonable to evaluate to what extent terms in a query obtained in the first step co-occur in the candidate documents. However, this measure does not always work well because common words are more likely to co-occur in the candidate and query. For instance, when having the two candidates related to "Obama" and "Yudhoyono" for the question "Who is the President of Indonesia?", the query "Obama AND President AND Indonesia" will extract more documents than the query "Yudhoyono AND President AND Indonesia" even though the correct answer is "Yudhoyono". Normalisation by the term frequency of each of the queries or evaluation of the conditional probability *P*(President|Yudhoyono) will help the QA system choose the correct answer. [20 pp. 90-91]

Clarke [41] presented Redundant Inverse Term Frequency (RITF) as

$$\mathrm{RITF}(t) = c_t \log(N/f_t),$$

where c_t is the number of the documents containing a term t, N is the total length of all documents and f_t is the term frequency of t. This metrics penalises the score of a query term when the term occurs repeatedly.

In the TREC QA track, participants were required to extract from the TREC corpus the strings which were assumed to contain an answer. Brill et al. made use of an external corpus in order to find answer candidates and do an "answer projection" [42 pp. 394].Their system then extracted similar expressions to the answer candidates from the TREC corpus [42].

Also, lexical similarity between a user's question and passages which are likely to answer the question is a sensible means to evaluate candidate answers. The similarity is, for

example, defined as the cosine of the angle between two document feature vectors, which may be formed from the TF-IDF scores of terms [43]. Word senses and parts of speech are thought to be other useful features for this purpose.

A beneficial feature of these measures is that they still do not depend on the question type of the user's question. It seems that the combination of these means and approaches using question types works complementarily. However, there is also a drawback that these systems require vast amounts of training data [20 pp. 92].

Moldovan et al. made use of a logic prover, COGEX [44], and their system performed best in the TREC QA track. In TREC-11, it obtained a confidence weighted score of 0.856 for answering factoid questions whilst the second ranked system only scored 0.691 [45], and in TREC-16, their system PowerAnswer 4 [46], based on COGEX, scored 70.6% precision for the factoid component with more than a 20% advantage over the second best system [47]. , quoted from [48], shows the architecture of COGEX. Overall, COGEX applies its English knowledge to analyse the representation of questions, collects world knowledge from a corpus like WordNet and abducts answers. If it fails to perform abduction, it will restart with looser conditions.

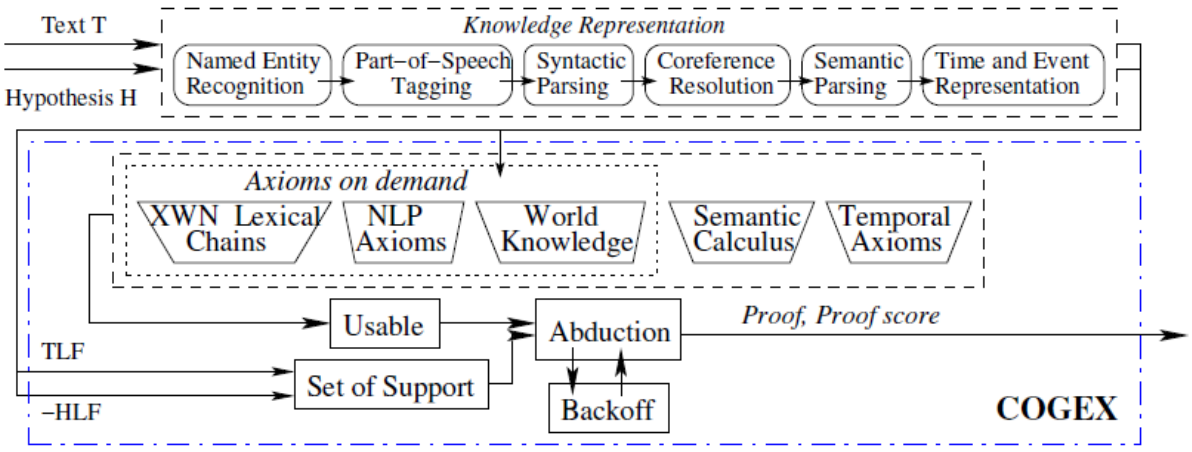

Figure 2.2. COGEX's Architecture [48]

A common method to do logical analysis is a resolution principle for knowledge written in a clausal form. Following the example in [49], in order to prove that Smith is an instance of animal, namely prove that Smith is an argument of atomic formula {ANIMAL(y)}, one shows a contradiction from knowledge with the negation {~ANIMAL(y)}. Now we have the two axioms, {MAN(Smith)} and {~MAN(x) or ANIMAL(x)}, which mean Smith is a man and x is not a man and therefore an animal, respectively. From the axioms, we can find {ANIMAL(Smith)} is correct. Here, the clausal form is proved because the fact contradicts the negation {~ANIMAL(y)}. As an example description format for knowledge, W3C have standardised the Resource Description Framework (RDF) [50]. Nowadays, there are many projects which are trying to build Knowledge Bases (KB) in a computer-friendly structure like RDF.

Some QA researchers address the answer evaluation problem with Statistical Machine Translation (SMT) models such as [28 pp.195-197]. Li and Manandhar [6] employed the model in order to extract terms representing the information needs of a question. The body of a question is expected to explicitly explain the information needs of the subject sentence in the question. Therefore, by applying a SMT model to the subject-body pairs of questions, the information needs of a short question can be inferred.

2.2.1.5 Evaluation Metrics

Factoid questions are relatively easy to evaluate as long as supervised test data are available. One of the most straightforward metrics is the precision of the QA system. In the case that the system is allowed to return multiple answers with a confidence rank rather than only a single answer, Mean Reciprocal Rank (MRR) is often used. MRR is the mean of the number of correct answers weighted by the reciprocal of the rank at which the answer is correct [20 pp. 99].

For questions requiring a list of answers, which were employed as the main tasks of the TREC-12 QA track, such as “which countries belong to the EU?”, the F measure is one of the most common metrics to measure the performance. The F measure is defined as

$$F = \left\{\alpha \frac{1}{R} + (1-\alpha)\frac{1}{P}\right\}^{-1} = \left[\frac{2RP}{R+P}\right]_{(\alpha=0.5)},$$

where R is a recall, a proportion of extracted correct answers out of all the correct ones and P is a precision, a proportion of correct answers in all the extracted answers. To maximise the F measure, balanced recall and precision are essential. As recall and precision are a trade-off, giving too many answers in order to produce a high precision lowers the recall and giving too few answers in order to produce a high recall ruins precision, and both these cases will therefore result in a low F measure [51 pp. 267-269].

TREC-13 employed the Confidence Weighted Score (CWS) to measure a series of questions sorted according to the rank of confidence. This score is the summation of the number of the correct answers for top- n questions divided by n, for $1 \leq n \leq (\#\ \text{of questions})$. In other words, as a system fails to answer more and more confident questions, the CWS diminishes.

In this thesis, the geometric mean of the highest rank of the correct answers was utilised for evaluation. In the case of using the arithmetic mean, the large figures of the rank will be dominant and therefore the difference of the first and the second place will not be shown. However, the geometric mean treats the second place as the twice of the first place. For example, the difference of the arithmetic means of 1 and 100, and 2 and 100 is only 0.5 whereas that of the geometric mean is about 4.14. This feature suits the evaluation of answer re-ranking.

2.2.2 Non-factoid Questions: Definition Questions

Non-factoid question answering systems have been very actively researched in recent years. From our point of view, the answer to factoid questions can be easily found by a commercial Web search engine because up-to-date search engines are powerful enough to retrieve relevant facts. The core value of question answering systems is therefore to extract answers of more complex questions, namely non-factoid questions. The TREC QA track did an evaluation of 'definition question' answering which contributed to the advance of non-factoid QA systems.

2.2.2.1 Early Definition Question Answering Systems

A well-known definition question answering system is FUNES (Figuring-out Unknown Nouns from English Sentences), which was created by Coates-Stephens in 1991 [52]. The

system attempts to extract the meaning of unknown nouns under the assumption that unknown nouns for the QA system are often also unknown for humans and accordingly the document includes the information about the noun. It utilises lexical knowledge, syntactic knowledge like appositives and manually crafted rules like “X is known as Y” [20 pp. 114]. Also, in 1992, Hearst made use of the relation between a hyponym and a hypernym like “X such as Y” [53].

At the end of the twentieth century, a statistical approach started being used in the study of Joho et al. [54]. They measured the certainty of manually crafted patterns and computed scores using the certainty as well as word frequency and position (because definition sentences are likely to appear at the very first part of articles) [20 pp. 118].

2.2.2.2 Procedure to Answer Definition Questions

The flow of definition QA is similar to that of factoid QA, described in Section 2.2.1, but each part is more difficult. In the first step, question analysis, systems have to clarify not only question types but also what is the target, or definiendum, in order to explain the definition [4 pp. 136]. When extracting answer candidates, it is reasonable to look for certain patterns as FUNES did (Section 2.2.2.1). Katz et al. defined 5 patterns: copular, appositive, occupation, verb and parenthesis patterns, in TREC-12 [55]. These patterns were able to be acquired through an automated process, shown in [12 and 35]. In the answer selection stage, a unique feature of this approach was to find expression styles which are likely to be definition statements. Both manually and automatically acquired patterns were applied to this process. Co-occurrence may be also a non-negligible attribute which a correct answer is expected to have. For definition questions, answer candidates may be chopped into terms, as bits of information representing the definition, when examining co-occurrence. As answer candidates are often only a part of the required information, the candidates should be merged with attention to redundancy. In the simplest way, highly ranked answer candidates are collected for merging [20 pp. 143].

Definition QA systems are often evaluated with the F measure. In the TREC-12 QA track, the F measure was applied with a weight regarding recall as 5 times more important than precision [56]. The lists of parts of definitions were prepared by the TREC assessors so as to compute F measure. There are also many other revised variants of the F measure such as Pyramid [57] and POURPRE [58].

2.2.2.3 BBN's Baseline System

BBN Technologies' baseline system, used in the TREC QA definition task, performed the best in TREC-12 scoring 0.493 [56]. Indeed, the reason for the high performance is that the baseline system output many words, up to 4000, as its answer and consequently obtained large recall. Under the F measure which places value equally on both precision and recall, the system recorded a low score, 0.205 [56], below the average of participants' scores. However, this baseline system may have some utility in building definition QA systems.

This system at first interpreted the targets of definition questions. Then, the system retrieved the top 1000 documents from data, resolved anaphorics, collected the sentences containing the targets from the retrieved documents and finally produced an answer within 4,000 characters. Sentences comprising 70% of redundant words were excluded [20 pp. 144-145].

2.2.2.4 BBN's System

BBN's challenge entry [41] selected answer sentences more carefully than the baseline system. The significant differences from the baseline were the introduction of kernel fact and question profile. Kernel fact is extracted by pattern matching and syntactic parse trees and ordered depending on how the kernel fact extracts. This extraction includes a method of resolution principal, using "predicate-argument structures" [59 pp. 101]. Question profile is a feature vector that shows which words co-occur with the target of the question in external corpora such as WordNet and Wikipedia. Then, an answer is produced by evaluating TF-IDF scores for kernel fact and question profile [20 pp. 145-147]. In TREC-12 [56], the system recorded the highest F score of 0.555 whereas the second-placed system recorded only 0.310.

2.2.2.5 QUALIFIER System and FDUQA System

QUALIFIER, QUestion Answering by LexIcal FabrIc and External Resources, developed at the National University of Singapore [60], is the system which achieved the second highest score of 0.473 [56] in TREC-12 without heavily relying on manually written rules. This system utilised snippets gathered from a Web search engine by using related words as keywords for retrieval. It collected a "positive set" of sentences which contained target terms or terms placed near the target terms, and a "negative set" of the sentences which are

the other parts of snippets. The answer was generated from sentences which are likely to appear in the "positive set" and unlikely to appear in the "negative set" [20 pp. 148-150].

FDUQA of Fudan University [61] is another system which not only performed well but was also constructed based on statistical approach. Although the implementation was different from QUALIFIER, the concept was fairly similar except that FDUQA makes use of online knowledge bases such as Wikipedia. If there were related entries in the knowledge bases to answer candidates, a part of the score was calculated as a similarity between the answer candidates and the entries [20 pp. 150-153].

2.2.2.6 PIQUANT System

Another interesting system was developed by IBM, called PIQUANT [62], even though the score in TREC-12 was only 0.177 [56]. The system divided a definition question into "auxiliary questions" [62 pp. 288], i.e. a set of factoid questions, and merged them in order to generate the answer. For example, answering a "Who is …?" format question, certain properties such as birth, college, marriage and death are expected to be required. These auxiliary questions were answered by one of seven agents which have different specialities. This system was quite unique and interesting, however, it could not cover a wide range of question types and thus, the score was low [56].

2.2.3 Non-factoid Questions: Why Questions

There is also much research which tries to answer reasons of a question starting with 'why', sometimes called why-question answering. This topic has not progressed very much because it is a tough task and why-question answering has not been actively addressed in major conferences. However, study of why-questions must play a crucial role for the advance of QA technology.

Why is answering why-questions difficult? First of all, why-questions often have multiple answers, which makes it hard to accomplish a high recall for the system. Secondly, the expressions of answers that are almost the same are lexically and syntactically distributed differently, whereas the expressions tend to be common in definition question answering. Finally, causalities form an endless chain. For example, a correct answer of the question "why is the sky blue?" is "blue light is scattered in the air more than the other colours", but

the questioner may want to know the reason behind this answer. The solution of these problems will make QA systems more human-like [63 pp. 1522].

2.2.3.1 Early Why-Question Answering Systems

Although practical why-question answering is a relatively new topic in NLP, the feature of causality has been studied, which is importantly related to why-question answering. CMACS (Causal Model Acquisition System) [64 as cited in 20 pp. 160-161] produced by Selfridge et al. in 1985 was one of the earliest systems. This system understood explanations (of engines) in natural language and answered a user's question based on automatically acquired knowledge from the users' explanations. In order to acquire the wisdom, the system utilised built-in knowledge (of engines) and lexical and syntactical knowledge.

2.2.3.2 Procedure to Answer Why-Questions

Current why-question answering systems often follow the ones for definition questions, aside from slight customisations. For instance, while appositives and some keywords like "such as" are utilised in a definition task, why-question answerers look for causal verbs "trigger" and causal links "because".

In the phase of question analysis, content words are used as keywords to search on the Internet instead of target words in a definition-question. Instances of content words are "king", "Egypt", "civilisation", "pyramid" and "construct" in the question "why did the Kings in Egyptian civilisation have pyramids constructed?" Causal keywords like "because" may be added to the set of keywords.

In the answer candidate extraction step, for example, fact sentences and reason sentences are recognised based on manually and/or automatically acquired patterns. Here, a fact sentence means one which contains facts related to a user's questions and a reason sentence means one which is likely to provide the reason. Lastly, answer candidates are assessed depending on how likely their expressions are to describe reasons as well as the lexical similarity to user's questions. It can be helpful to consider term pairs having causalities. In the case that a question includes the word "die", the answer is expected to include the cause of death, such as a heart attack.

To evaluate why-question answering systems, MRR and top-n success precision are commonly employed. Metrics used in definition question answering systems like F measure may also work well [20 pp.169-170].

2.2.3.3 Existing Why-Question Answering Systems

Similar to other types of QA systems, manually crafted rules were utilised in early why-QA systems. Girju proposed an approach to acquire causation relations via manually defined causal verbs in 2003 [65]. Morooka prepared positive patterns and negative patterns based on cue words so as to differentiate causal words [66 as cited in 20 pp. 179-180]. The approach utilised by Shibusawa et al. was to add either "naze" or "doushite", Japanese interrogatives "why", to queries and to weight each causal patterns according to the degrees of confidence which were manually inferred. This system, called RE:Why, achieved a MRR score of 0.552 and a precision of 37.1% for the top 10 answers [63].

Although manual acquisition of rules is simple and powerful for a specific domain, it is impossible to cover all patterns of natural language. NAZEQA, developed by Higashinaka and Isozaki in 2008 [67 pp. 143] is a system which acquires causal expressions automatically. They used the EDR corpus, the entries of which included CAUSE tags indicating causal relations. As a result of analysing sentences with CAUSE labels syntactically and semantically, NAZEQA achieved a top-5 MRR score of 0.309 [68]. Note that, although this score seems lower than RE:Why system's, it is not comparable due to the different text data sets.

Another interesting method was proposed by Verberne. The main idea of her system is to compute "term overlaps" with taking into account "syntactic functions and categories" [69]. This is quite a simple and generally applicable idea, and the system achieved good scores, a top-150 MRR of 0.34 and a top-10 precision of 57%. This compared well with the MRR score of the system without structural overlap information, which was only 0.25 [70 pp. 240].

A system proposed by Tamura et al. rephrased a question into words which are more likely to occur in corpora [70]. Their system retrieved snippets from Google by using the rephrased words and output the snippets which include the words. The simple method worked surprisingly well, with an MRR score of 0.409 and 54% precision for top-10

outputs. From my point of view, this method may suit Japanese more than English due to the language's attributes.

Oh et al. reported that sentiment analysis improved the performance of why-QA systems [71]. They assumed that "if something undesirable happens, the reason is often also something undesirable" [71 pp. 368], and the converse for the event of something desirable happening. For instance, the question "why does rickets occur in children?" contains a negative word "rickets" and the answer candidate "deficiency of vitamin D can cause rickets" contain another negative word "deficiency" aside from "rickets" [71 pp. 377]. Consequently, the system can judge that the question and the candidate are sentimentally related. The precision score was 0.336, which was 0.028 higher than the system without sentiment analysis. Indeed, since this sentiment analysis is fairly shallow, deeper sentiment analysis can be expected to improve a why-QA systems' performance more.

2.2.4 Other Systems for Non-factoid Question Answering

2.2.4.1 How-to-Question Answering

Similar to why-QA, how-to-QA represents another challenging task. This task is relatively undeveloped amongst QA; however, it can be thought that there is large demand. Asanoma et al. [72] reported that itemised lists would be a key fact to answer questions asking for a certain procedure. They also showed weaker correlation between tables and such questions. This is an appropriate approach to apply to websites because HTML documents are structured by tags.

2.2.4.2 Opinion Question Answering

In the Text Analysis Conference (TAC) 2008, participants dealt with an opinion QA task. The task was to answer two types of questions, rigid and squishy questions, by using a blog corpus. For example, a rigid question was "who likes Mythbuster's?" and a squishy one was "why do people like Mythbuster's?" [73].

Li et al. developed QUANTA [74], which was placed first by the metrics of the conference. This system incorporated sentiment analysis (to classify opinions into positive or negative) as well as semantic-oriented parse trees. It used some patterns such as "opinion about X" and also opinion dictionaries in order to detect terms likely to refer opinions. In addition, it

divided rigid questions into ones about a person holding an opinion and the others to apply distinct algorithms.

Li et al. improved the system above by using a Markov random walk model [74]. Inspired by the PageRank algorithm [75], they developed a new system which builds a graph of importance of each sentence based on the similarity to each of another sentence as well as the similarity between the sentence and a question. The graph is then revised based on the relations to sentiment words. They also proposed that relations among answer candidates should be taken into account because redundant candidates are more likely to be an important opinion. These measures contributed to the 20% higher F score than QUANTA in TAC 2008.

2.2.5 Recognising Textual Entailment

Recognising Textual Entailment (RTE) is also a frequently researched topic nowadays. When one says "T entails H", where T is a text and H is a hypothesis, a human can infer from T that H is most probably true [77]. In RTE, researchers are required to take into account not only entailment but also contradiction and irrelevance. Thanks to the contribution of TAC, datasets for RTE came to be available and this has accelerated the study of RTE. QA systems are hitting upper bounds due to the limitations of superficial similarity models. This is despite much room for improvement in their performance. On the other hand, RTE enables researchers to build a syntactically, semantically more robust model. In the QA4MRE track of CLEF 2011, the top scoring team achieved a 56% higher score than the runners up [78]. The fact that they were the only one to utilise RTE among the teams who used the English corpus truly tells us that RTE is important.

Machine learning approaches have been widely applied to RTE studies because hand craft rules have serious problems: one is the coverage of the rules and another is that rewriting a rule often affects the other rules [79]. Many RTE systems have rewriting rules, namely ground and first-order rewriting rules. The difference is that ground ones do not apply variables but first-order ones do. Ground rewriting rules include rules such as "The sun emits UVA rays" -> "Tanning can expose you to health risks" as well as "Oswald killed JFK" -> "JFK died", while first-order rewrite rules are defined such as "X killed Y" -> "Y died" [79 pp. 552-553]. In a machine learning process, these rules are acquired by analysing texts lexically and syntactically [79].

To obtain semantically reasonable output, logical inference should be taken into account. Bos and Markert incorporated a theorem prover (as shown in 2.2.1.4), which tries to prove an input theorem, and a model builder, which inversely looks for the negation of input [80]. Either of these processes will detect whether or not T entails H. In order to derive the entailment, the theorem prover attempts to prove two formulae which say: “T implies H” and “T+H are inconsistent” [80 pp. 630]. The former shows entailment and the latter no entailment. The theorem prover has background knowledge to prove the formulae: manually-crafted generic knowledge, lexical knowledge and geographical knowledge. In the case that the prover cannot answer the input, the system assumes that T probably entails H if T+H are substantially informative against T [80].

2.3 Footing of the proposed system among QA systems

This research is based on a topic model with a corpus extracted from a SQA forum. This section describes directly related work in order to clarify the footing of this study in the field of QA.

2.3.1 Topic models

It is intuitive that when the topics of documents are the same, the topic distributions of these documents will be similar to each other. A topic model is one that assumes each document has a topic distribution associated with the topics of the words in the document. The theoretical explanation is detailed in the chapter 3.

The model is not only used for the classification of documents, but also used to bridge the lexical chasm between words in different contexts. Celikyilmaz, Hakkani-Tur and Tur made use of topic models in order to compute the similarity between a question and a candidate answer passage with the metrics shown in 4.3 [81]. The above-mentioned research by Li and Manandhar followed their idea to apply it to a question recommendation system [6].

Liu et al. made use of a topic model to predict best answerers in a social QA forum. They took answers by a user as a document and analysed the topics for which the user tends to answer. This idea is also useful to address the lexical chasm problem.

2.3.2 Social Question Answering Forum

One of the thorniest problems in QA research is the difficulty in preparing a corpus. Thus, social question answering (SQA) forums [2, 3, 4, 5, 6, 7, 8, 9 and 10], such as Yahoo! Answers, come to be important these days. Note that SQA is also called community-based question answering (CQA). SQA forums are rich in instances of sets of questions and answers which are truly open to all domains regardless of language. Not only do these benefits exist, but the websites also have useful features to infer the quality of answers. For example, a questioner (and often visitors of the QA thread) can vote answers as to whether or not they are useful, and thus a QA system can track excellent answerers who get a large number of positive votes. It is another benefit that researchers can utilise the resources for free. On the other hand, these instances are often ill-formed; and include word omissions, implications and even incorrect or irrelevant information. Yuanjie Liu et al. and Yandong Liu et al. independently studied the statistics of features of SQA services [3 and 4]. Although it is essential to have noise reduction and information complementation, the use of QA communities will be helpful to realise a sophisticated QA system.

Xue et al. proposed a retrieval model for QA community archives [5]. The model was based on a statistical machine translation (SMT) model, where the system translates questions into answers regarded as a monolingual corpus and vice versa. By retrieving information from both questions and answers with the model, the top-10 precision reached 0.3, while that of a simple Okapi/BM25 model was 0.21. Li and Manandhar also incorporated a SMT model to infer information needs of a question by translating the title of a question into the content of the question [6]. They suggested that the translation of a new question title expresses information needs and this idea can be exploited in recommending answers.

Wang, Ming and Chua incorporated “syntactic tree matching” [10], as explained in Section 2.2.1.3. This model improved the top-10 MAP score from 81.61% to 85.67% as compared to the original kernel tree function model.

2.3.2.1 Use of Question and Answer Pairs as Training Set

A few researchers have exploited question and answer pairs retrieved from such communities to train their QA systems so as to then utilise the system with another corpus. Indeed, the preparation of Q&A instances is generally a tough task. It can be thought that a

reason why the TREC QA track has boosted the research of QA was that TREC had prepared a sufficient amount of Q&A instances for training. To the best of my knowledge, there are only two groups who used SQA-based Q&A instances to analyse documents from SQA forums such as news articles. Many researchers are interested in other challenges, e.g. answer quality ranking and question recommendation.

In 2008, Mori, Okubo and Ishioroshi [7] made use of instances of Q&A pairs in order to match the writing styles with answer candidates, instead of question types. The benefits were that the system did not have to employ a manually defined question-type classifier and that it was able to obtain more specific classes for non-factoid questions than primitive question types like definition-type and why-type. Wu and Kawai made similar use of Q&A pairs from SQA websites in 2010 [8]. They developed a system which extracts answers from Chinese newspapers by utilising the "Question-Type-Specific Method" (QTSM), in which question-types are acquired automatically through supervised CRF model training. They reported that their system based on QTSM got a higher F3 score than a system based on a statistical machine translation model.

Both groups simply used "best answers" of questions from SQA websites and discarded the others although the latter research also examined a system which removes noisy answers and this enhanced its F3 score. It may therefore be fruitful to select only high quality answer candidates for the training. There is much research which has investigated how to distinguish high quality answers like Surdeanu, Ciaramita and Zaragoza [9]. Proper answer selection for training sets may be helpful to optimise the answer ranking algorithm.

2.3.3 Query Expansion

Query expansion is an approach used to address problems associated with synonyms. For example, when a user searches on an information retrieval system about 'tutor', some documents with important information may contain 'instructor' instead of 'tutor'. In this case, the user fails to obtain the information in spite of the fact that it is very easy for humans to relate the two words. To avoid this problem, one can rephrase a word into its synonyms and search for the information with all of the synonyms including the original word.

Research proposing the combination of a topic model and a query expansion model is described by Park and Ramanohanarao [83]. They used PLSA (shown in 3.1) and similarity metrics based on Okapi/BM25 (shown in 2.2.1.2). Their approach offers an advantage in terms of memory efficiency because of the lower dimensions of the topics as compared to that of words and documents.

3 Topic Model

The idea of a topic model is that each document and each term within contain information about their topics based on the empirical knowledge that related terms tend to co-occur in the same document. For instance, the terms 'doctor' and 'medicine' are likely to appear in the same document and 'tree' and 'forest' are also likely whereas 'doctor' and 'tree', 'medicine' and 'forest' are unlikely to co-occur. It is then natural to assume that 'doctor' and 'medicine' will share a medical topic, and 'tree' and 'forest' will share a nature topic.

A topic model is advantageous for information retrieval because it can take into account synonyms and polysemes. Also, the model is utilised for document clustering, cross-language information retrieval, and many more. By incorporating the ability to find synonyms and polysemes, query expansion has been performed in several pieces of research, including this research.

This chapter summarises the theoretical background of topic models. This provides an understanding of how the system infers the semantic features of words over the question-answer document aligned corpus.

3.1 Topic Models

An early topic model was based on the Probabilistic Latent Semantic Analysis (PLSA, or Probabilistic Latent Semantic Indexing, PLSI) model proposed by Hofmann et al. in 1998 [84]. PLSA incorporates an EM algorithm over multinomially-distributed variables whereas LSA [85] incorporates a linear algebraic approach. In a PLSA model, for a document $\boldsymbol{d}$ and a term t, $P(\boldsymbol{d}, t)$ is calculated via hidden variables $\boldsymbol{z}$ as:

$$P(\boldsymbol{d}, t) = \sum_{z} P(z)P(\boldsymbol{d}|z)P(t|z) = P(\boldsymbol{d}) \sum_{z} P(z|\boldsymbol{d})P(t|z).$$

Then, the hidden variables will express the topics of the documents and terms. A graphical representation of the PLSA model is shown in Figure 3.1.

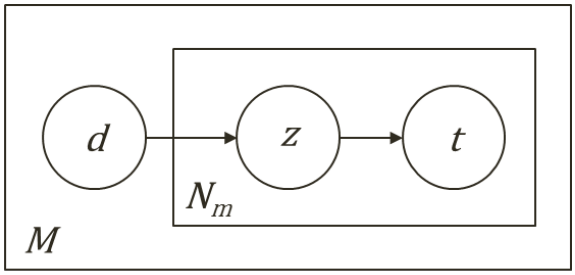

Figure 3.1. A diagram of a PLSA model

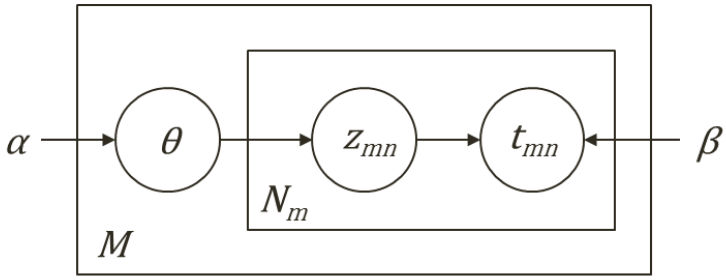

Figure 3.2. A diagram of an unsmoothed LDA model

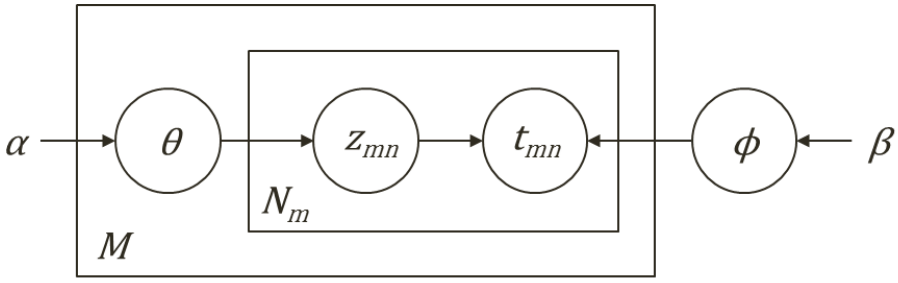

Figure 3.3. A diagram of a smoothed LDA model

Latent Dirichlet Allocation (LDA) proposed by Blei et al. in 2003 [13] is another topic model which is commonly utilised in NLP. In LDA, topic distributions and word distributions (or term distributions) are sampled from Dirichlet distributions, which are conjugate priors of multinomial distributions. Employing Dirichlet distributions has advantages, for example, when dealing with unknown terms and when working on a small corpus. The difference between an LDA model and a PLSA model is illustrated graphically by comparing Figure 3.1 and Figure 3.2. In this chapter, a basic LDA model [13] will be introduced first and it will then be expanded into a Bi- LDA model [1].

3.2 LDA Models

Graphical models of LDA are shown in Figure 3.2 and Figure 3.3. In the smoothed LDA model, a set of documents is generated through the following procedure.

1. Per-document topic distributions $\boldsymbol{\theta}$ and per-term topic distributions $\boldsymbol{\phi}$ are sampled from Dirichlet distributions with Dirichlet priors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively.
2. For each document m:
    a. For each term n:
        i. A topic z_{mn} is sampled according to the multinomial distribution for the prior $\boldsymbol{\theta}_m$.
        ii. A term is sampled according to the multinomial distribution for the prior $\boldsymbol{\phi}$ corresponding to the topic z_{mn}.

Since per-document topic distributions and per-topic term distributions are taken as Dirichlet distributions, which are conjugate priors of multinomial distributions, an LDA model can take into account terms which have not appeared in the corpus while a PLSA model [81] cannot.

An LDA model is a generative model in that documents are generated with per-document topic distributions and per-topic term distributions. The generation probability is:

$$P(\boldsymbol{D}|\boldsymbol{\alpha},\boldsymbol{\beta}) = \prod_{m=1}^{M} \int P(\boldsymbol{\theta}_m|\boldsymbol{\alpha}) \left\{ \prod_{n=1}^{N_m} \sum_{k=1}^{K} P(z_{mn}=k|\boldsymbol{\theta}_m) P(t_{mn}|z_{mn},\boldsymbol{\beta}) \right\} d\boldsymbol{\theta}_m,$$

where $\boldsymbol{D}$ is the set of documents, $\boldsymbol{\alpha}$ is a Dirichlet prior, $\boldsymbol{\beta}$ is a multinomial prior, M and K are the numbers of documents and topics, respectively, N_m is the number of terms in the m-th document, $\boldsymbol{\theta}_m$ is a per-document topic distribution for the m-th document, and z_{mn} and t_{mn} are the topic and the term, respectively, of the n-th term in the m-th document. Then, Bayesian methodology enables us to infer per-document topic distributions and per-topic term distributions from the observed terms in the set of documents. Blei et al. originally proposed an LDA model with Variational Bayesian (VB) inference in 2003 [13]. Griffiths and Steyvers [14] reported a Collapsed Gibbs Sampling (CGS) approach performed better than the VB approach in terms of speed and perplexity because the VB approach is computationally expensive and derived under the inaccurate assumption of

mutual independence between topics, per-document topic distributions and per-topic term distributions [15]. Teh et al. improved the VB inference and called it Collapsed Variational Bayesian (CVB) inference [15]. Asuncion et al. reported a variant of CVB approaches performed quicker and with higher precision than both the VB approach and the CGS approach [16].

3.2.1 Variational Bayesian Inference

The original approach of LDA proposed by Blei, Ng and Jordan in 2003 [13] is classified under the VB approach. The generation probability is computationally intractable due to the coupling of two latent variables $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, for example in the unsmoothed model shown in Figure 3.2. However, the generation probability is required in order to compute the following posterior probability:

$$P(\boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{T}; \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{P(\boldsymbol{\theta}, \boldsymbol{Z}, \boldsymbol{T}|\boldsymbol{\alpha}, \boldsymbol{\beta})}{P(\boldsymbol{T}|\boldsymbol{\alpha}, \boldsymbol{\beta})}$$

where $\boldsymbol{T}$ and $\boldsymbol{Z}$ are the series of terms and topics, respectively, for each document. To solve the problem, the following approximation is incorporated instead of $P(\boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{T}; \boldsymbol{\alpha}, \boldsymbol{\beta})$:

$$q(\boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{\mu}, \boldsymbol{\gamma}) = \prod_{m=1}^{M} q(\boldsymbol{\theta}_m|\boldsymbol{\mu}_m) \prod_{n=1}^{N_m} q(z_{mn}|\boldsymbol{\gamma}_{mn})$$

where $q(\boldsymbol{\theta}_m|\boldsymbol{\mu}_m)$ obeys the Dirichlet distribution with the prior $\boldsymbol{\mu}$ and $q(z_{mn}|\boldsymbol{\gamma}_{mn})$ obeys the multinomial distribution with the prior $\boldsymbol{\gamma}_{mn}$. In this approximation, the dependency between topics $\boldsymbol{Z}$ and per-document topic distributions $\boldsymbol{\theta}$ disappears. As a result, the VB inference tends to be less accurate than the CVB and CGS inference which are introduced the coming sections. Then, these parameters are optimised via minimising the Kullback-Leibler (KL) divergence between the approximated probability and the true probability:

$$(\boldsymbol{\mu}^*, \boldsymbol{\gamma}^*) = \underset{(\boldsymbol{\mu}, \boldsymbol{\phi})}{\operatorname{argmin}} D_{KL}\big(q(\boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{\mu}, \boldsymbol{\gamma})||P(\boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{T}; \boldsymbol{\alpha}, \boldsymbol{\beta})\big).$$

On the other hand, the loss function so as to optimise $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is:

$$l(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{m=1}^{M} \log P(\boldsymbol{t}_m|\boldsymbol{\alpha}, \boldsymbol{\beta}).$$

The approximation enables this loss function to be tractable via Jensen's inequality. For a concave function $f(y)$, such as $\log y$, Jensen's inequality is [86]:

$$E[f(y)] \leq f(E[y]).$$

Therefore,

$$\begin{aligned}\log P(\boldsymbol{t}|\boldsymbol{\alpha},\boldsymbol{\beta}) &= \log \int \sum_{k=1}^{K} P(\boldsymbol{\theta},\boldsymbol{z},\boldsymbol{t}|\boldsymbol{\alpha},\boldsymbol{\beta})\, d\boldsymbol{\theta} \\ &= \log \int \sum_{k=1}^{K} \frac{P(\boldsymbol{\theta},\boldsymbol{z},\boldsymbol{t}|\boldsymbol{\alpha},\boldsymbol{\beta}) q(\boldsymbol{\theta},\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\gamma})}{q(\boldsymbol{\theta},\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\gamma})} d\boldsymbol{\theta} \\ &\geq E_q[\log P(\boldsymbol{\theta},\boldsymbol{z},\boldsymbol{t}|\boldsymbol{\alpha},\boldsymbol{\beta})] - E_q[\log q(\boldsymbol{\theta},\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\gamma})]\end{aligned}$$

where the subscripts m, indicating the index of the document, are omitted for $\boldsymbol{t}_m$, $\boldsymbol{z}_m$ and $\boldsymbol{\theta}_m$. Since the difference between the left-hand side and the right-hand side of this inequality corresponds to the KL divergence between the variational posterior probability and the true posterior probability, then according to Blei et al optimisation via KL divergence is justified. Namely, letting the right-hand side of the inequality be equal to $L(\boldsymbol{\mu},\boldsymbol{\gamma};\boldsymbol{\alpha},\boldsymbol{\beta})$,

$$\log P(\boldsymbol{t}|\boldsymbol{\alpha},\boldsymbol{\beta}) = L(\boldsymbol{\mu},\boldsymbol{\gamma};\boldsymbol{\alpha},\boldsymbol{\beta}) + D_{KL}\big(q(\boldsymbol{\theta},\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\gamma})||P(\boldsymbol{\theta},\boldsymbol{z}|\boldsymbol{t};\boldsymbol{\alpha},\boldsymbol{\beta})\big)$$

By formulating $L(\boldsymbol{\mu},\boldsymbol{\gamma};\boldsymbol{\alpha},\boldsymbol{\beta})$, the parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are computed via the method of Lagrange multipliers with the constraint that the summation of probability is unity, and the other parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\mu}$ are computed by differentiating the loss functions. The Newton-Raphson algorithm is used for $\boldsymbol{\alpha}$ and the obtained formulae are then:

$$\begin{aligned}&\boldsymbol{\alpha}^* = \boldsymbol{\alpha} - H(\boldsymbol{\alpha})^{-1} g(\boldsymbol{\alpha}), \\ &{\beta_{kv}}^* \propto \sum_{m=1}^{M} \sum_{n=1}^{N_m} \gamma_{mnk} \cdot u(t_{mn} = v)\,, \\ &{\mu_{mk}}^* = \alpha_i + \sum_{n=1}^{N_m} \gamma_{mnk}\,, \\ &{\gamma_{mnk}}^* \propto \beta_{kt_{mn}} \cdot \exp\big(E_q[\log \theta_{mk}\,|\boldsymbol{\mu}_m]\big),\end{aligned}$$

where $H(\boldsymbol{\alpha})$ and $g(\boldsymbol{\alpha})$ are the Hessian matrix and gradient, respectively, k denotes the index of topics, v denotes the index of the term in the vocabulary, and $u(t_{mn} = v)$ is 1 if $t_{mn} = v$, 0 otherwise. The expectation $E_q[\log \theta_{mk} | \boldsymbol{\mu}_m]$ is computed as:

$$E_q[\log \theta_{mk} | \boldsymbol{\mu}_m] = \Psi(\mu_{mk}) - \Psi\left(\sum_{i=1}^{K} \mu_{mi}\right)$$

where Ψ is the first derivative of the log gamma function computed via Taylor approximations [87].

To summarise, LDA inference is performed via the following variational EM algorithm:

1. Initialise $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.
2. (E-Step) Optimise $\boldsymbol{\mu}$ and $\boldsymbol{\gamma}$ according to the current $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.
3. (M-Step) Compute $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.
4. Repeat 2 to 3 until convergence.

For the m-th document, $\boldsymbol{\mu}_m$ and $\boldsymbol{\gamma}_m$ are inferred with another variational approach as follows:

1. Initialise $\gamma_{mnk}^{0} = 1/K$ for all k and n
2. Initialise $\mu_{mk}^{0} = \alpha_k + N/K$ for all k
3. For $n = 1$ to N_m
    a. For $k = 1$ to K
        i. $\gamma_{mnk}^{j+1} = \beta_{kt_{mn}} \cdot \exp\left(\Psi\left(\mu_{mk}^{j}\right)\right)$
    b. Normalise $\boldsymbol{\gamma}_{mn}^{j+1}$ to sum to 1.
4. $\boldsymbol{\mu}_m^{j+1} = \boldsymbol{\alpha} + \sum_{n=1}^{N_m} \boldsymbol{\gamma}_{mn}^{j+1}$
5. Repeat 3 to 4 until convergence

3.2.2 Collapsed Variational Bayesian Inference

This approach offers advantages in computing speed and convergence perplexity although it can suffer a local optimum problem unlike the CGS approach. The convergence perplexity will also be lower than the basic VB approach since the CVB approach does not employ the independence assumption among topics, per-document topic distributions and

per-topic term distributions [15]. Thus, the approximation of the posterior probability is redefined:

$$q(\boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{\gamma}) = q(\boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{Z}) \prod_{m=1}^{M} \prod_{n=1}^{N_m} q(z_{mn}|\boldsymbol{\gamma}_{mn})$$

First of all, the variational free energy of $q(\boldsymbol{Z})q(\boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{Z})$ is:

$$\mathcal{F}\big(q(\boldsymbol{Z})q(\boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{Z})\big) = E_{q(\boldsymbol{Z})q(\boldsymbol{\theta},\boldsymbol{\phi}|\boldsymbol{Z})}[-\log P(\boldsymbol{T}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})] - H\big(q(\boldsymbol{Z})q(\boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{Z})\big)$$

$$= E_{q(\boldsymbol{Z})}[E_{q(\boldsymbol{\theta},\boldsymbol{\phi}|\boldsymbol{Z})}[-\log P(\boldsymbol{T}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})] - H\big(q(\boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{Z})\big)] - H\big(q(\boldsymbol{Z})\big)$$

By minimising the energy in terms of $q(\boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{Z})$ and $\boldsymbol{\gamma}$ step by step, the following $\boldsymbol{\gamma}$ is obtained:

$$\gamma_{mnk} = q(z_{mn} = k) = \frac{\exp\left(E_{q(Z^{\neg mn})}[P(\boldsymbol{T}, \boldsymbol{Z}^{\neg mn}, z_{mn} = k|\boldsymbol{\alpha}, \boldsymbol{\beta})]\right)}{\sum_{i=1}^{K} \exp\left(E_{q(Z^{\neg mn})}[P(\boldsymbol{T}, \boldsymbol{Z}^{\neg mn}, z_{mn} = i|\boldsymbol{\alpha}, \boldsymbol{\beta})]\right)}$$

$$= \frac{\exp\left(E_{q(Z^{\neg mn})}[\log(\alpha_k + n_{nk\cdot}^{\neg mn}) + \log\big(\beta_{t_{mn}} + n_{\cdot kt_{mn}}^{\neg mn}\big) - \log(\sum_{p=1}^{V} \beta_p + n_{\cdot k\cdot}^{\neg mn})]\right)}{\sum_{i=1}^{K} \exp\left(E_{q(Z^{\neg mn})}[\log\big(\alpha_i + n_{ni\cdot}^{\neg mn}\big) + \log\big(\beta_{t_{mn}} + n_{\cdot it_{mn}}^{\neg mn}\big) - \log(\sum_{p=1}^{V} \beta_p + n_{\cdot i\cdot}^{\neg mn})]\right)}$$

where n_{nkv} denotes the number of documents that the n-th term in the document is v and assumed as topic k,,and V is the number of unique terms in the corpus.. The symbol '$\cdot$' denotes the accumulation of the elements, and $\neg mn$ denotes the exception of the n-th term in the m-th document. γ_{mnk} is expanded by applying $\log\frac{\Gamma(\eta+n)}{\Gamma(\eta)} = \sum_{l=0}^{n-1} \log(\eta + l)$ for positive real numbers η and positive integers n. Indeed, this formula is rather computationally expensive for practical use. A Gaussian approximation helps to significantly reduce the computational costs with high accuracy. Then the mean and variance become:

$$E_q[n_{nk\cdot}^{\neg mn}] = \sum_{i \neq m} \gamma_{ink}$$

$$\mathrm{Var}_q[n_{nk\cdot}^{\neg mn}] = \sum_{i \neq m} \gamma_{ink}(1 - \gamma_{ink}).$$

Also, a second-order Taylor expansion reduces the computation costs while keeping the inference accurate:

$$E_q[\log(\alpha_k + n_{nk\cdot}^{\neg mn})] \sim \log\left(\alpha_k + E_q[n_{nk\cdot}^{\neg mn}]\right) - \frac{\mathrm{Var}_q[n_{nk\cdot}^{\neg mn}]}{2\left(\alpha_k + E_q[n_{nk\cdot}^{\neg mn}]\right)^2}.$$

Thus, using uniform $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, namely $\alpha_k = \alpha$ and $\beta_v = \beta$ for all k and v, $\boldsymbol{\gamma}$ can be calculated as:

$$\gamma_{mnk} \propto \left(\alpha + E_q[n_{nk\cdot}^{\neg mn}]\right) \frac{\beta + E_q\left[n_{\cdot kt_{mn}}^{\neg mn}\right]}{V\beta + E_q[n_{\cdot k\cdot}^{\neg mn}]}$$
$$\cdot \exp\left(-\frac{\mathrm{Var}_q[n_{nk\cdot}^{\neg mn}]}{2\left(\alpha + E_q[n_{nk\cdot}^{\neg mn}]\right)^2} - \frac{\mathrm{Var}_q\left[n_{\cdot kt_{mn}}^{\neg mn}\right]}{2\left(\beta + E_q\left[n_{\cdot kt_{mn}}^{\neg mn}\right]\right)^2} + \frac{\mathrm{Var}_q[n_{\cdot k\cdot}^{\neg mn}]}{2(V\beta + E_q[n_{\cdot k\cdot}^{\neg mn}])^2}\right).$$

Iteratively updating $\boldsymbol{\gamma}$, $n_{nk\cdot}$, $n_{\cdot kt_{mn}}$ and $n_{\cdot k\cdot}$ enable us to infer the per-document topic distributions and the per-topic term distributions accurately.

Asuncion et al. [16] suggested applying a zero-order Taylor expansion to this update formula to give a further approximation. Namely,

$$\gamma_{mnk} \propto \left(\alpha + E_q[n_{nk\cdot}^{\neg mn}]\right) \frac{\beta + E_q\left[n_{\cdot kt_{mn}}^{\neg mn}\right]}{V\beta + E_q[n_{\cdot k\cdot}^{\neg mn}]}.$$

They called the model with this significantly simplified formula CVB0. CVB0 works surprisingly well despite the simplicity and will bring quicker inference than a CGS model, and more accurate results than a VB model. The speed is a benefit of not requiring a sampling, which is basically computationally expensive. Moreover, if the same values are assigned to $\boldsymbol{\gamma}$ for the same terms in a document, there is no need to redundantly compute the $\boldsymbol{\gamma}$ because obviously all of the $\boldsymbol{\gamma}$ will stay equal after updating. On the other hand, the accuracy comes from the smaller approximation in terms of the independence of topics $\boldsymbol{z}$ and per-document topic distributions $\widehat{\boldsymbol{\theta}}$, as mentioned above.

3.2.3 Collapsed Gibbs Sampling

CGS approaches [14] are commonly incorporated in various LDA models because they tend to be quicker and more precise in inference and easier to implement than the vanilla

VB model. As noted in the previous section, the accuracy comes from considering the dependency among topics, per-document topic distributions and per-topic term distributions. The advantage of converging to the global optimum is due to the sampling approach.

The summary of the algorithm for CGS is as follows:

1. Set Dirichlet hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$
2. Initialise each term with a topic
3. For each document m ($1 \leq m \leq M$)
    i. For each term t_{mn} ($1 \leq n \leq N_m$)
        - Sample the topic z_{mn} for t_{mn}
4. Return to 3 until convergence

Topics are sampled according to the following probabilities for uniform $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$:

$$z_{mn} \sim (\alpha + c_{mk\cdot}^{\neg mn}) \frac{\beta + c_{\cdot k t_{mn}}^{\neg mn}}{V\beta + c_{\cdot k\cdot}^{\neg mn}},$$

where c_{mkv} is the count of the term v which is assigned to the topic k in the m-th document, t_{mn} is the n-th term in the m-th document, and V is the number of the kind of terms in the vocabulary. The symbol '$\cdot$' means the accumulation of the corresponding elements and the superscript $\neg mn$ means the exception of the term t_{mn}. For example, $c_{mk\cdot}^{\neg mn}$ means the count of the terms assigned to the topic k in the m-th document regardless of the kind of terms except for the term t_{mn}.

The probabilities are introduced by integrating the posterior probability $P(z_{mn} = k|\boldsymbol{Z}^{\neg mn}, \boldsymbol{T}; \boldsymbol{\alpha}, \boldsymbol{\beta})$ with per-document topic distributions $\boldsymbol{\theta}$ and per-topic term distributions $\boldsymbol{\phi}$. Namely,

$$\begin{aligned} &P(z_{mn} = k|\boldsymbol{Z}^{\neg mn}, \boldsymbol{T}; \boldsymbol{\alpha}, \boldsymbol{\beta}) \\ &\quad = \frac{P(z_{mn} = k, t_{mn}|\boldsymbol{Z}^{\neg mn}, \boldsymbol{T}^{\neg mn}; \boldsymbol{\alpha}, \boldsymbol{\beta})}{P(t_{mn}|\boldsymbol{Z}^{\neg mn}, \boldsymbol{T}^{\neg mn}; \boldsymbol{\alpha}, \boldsymbol{\beta})} \\ &\quad \propto \int_{\boldsymbol{\theta}} \int_{\boldsymbol{\phi}} P(z_{mn} = k, t_{mn}, \boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{Z}^{\neg mn}, \boldsymbol{T}^{\neg mn}; \boldsymbol{\alpha}, \boldsymbol{\beta}) d\boldsymbol{\phi} d\boldsymbol{\theta} \end{aligned}$$

$$= \int_{\boldsymbol{\theta}}\int_{\boldsymbol{\phi}} P(z_{mn}=k,\boldsymbol{\theta},\boldsymbol{\phi}|\boldsymbol{Z}^{\neg mn},\boldsymbol{T}^{\neg mn};\boldsymbol{\alpha},\boldsymbol{\beta})$$

$$\times P(t_{mn},\boldsymbol{\theta},\boldsymbol{\phi}|z_{mn}=k,\boldsymbol{Z}^{\neg mn},\boldsymbol{T}^{\neg mn};\boldsymbol{\alpha},\boldsymbol{\beta})d\boldsymbol{\phi}d\boldsymbol{\theta}$$

$$= \int_{\boldsymbol{\theta}} P(z_{mn}=k,\boldsymbol{\theta}|\boldsymbol{Z}^{\neg mn};\boldsymbol{\alpha})d\boldsymbol{\theta} \times \int_{\boldsymbol{\phi}} P(t_{mn},\boldsymbol{\phi}|z_{mn}=k,\boldsymbol{T}^{\neg mn};\boldsymbol{\beta})d\boldsymbol{\phi}$$

$$\propto \int_{\boldsymbol{\theta}_m} P(z_{mn}=k|\boldsymbol{\theta}_m)P(\boldsymbol{\theta}_m|\boldsymbol{Z}^{\neg mn};\boldsymbol{\alpha})d\boldsymbol{\theta}_m$$

$$\times \int_{\boldsymbol{\phi}_k} P(t_{mn}|z_{mn}=k,\boldsymbol{\phi}_k)P(\boldsymbol{\phi}_k|\boldsymbol{T}^{\neg mn};\boldsymbol{\beta})d\boldsymbol{\phi}_k$$

$$\propto \int_{\theta_{mk}} \theta_{mk}P(\theta_{mk}|\boldsymbol{Z}^{\neg mn};\boldsymbol{\alpha})d\theta_{mk} \times \int_{\phi_{kt_{mn}}} \phi_{kt_{mn}}P\left(\phi_{kt_{mn}}|\boldsymbol{T}^{\neg mn};\boldsymbol{\beta}\right)d\phi_{kt_{mn}}$$

$$= E[\theta_{mk}|\boldsymbol{Z}^{\neg mn};\boldsymbol{\alpha}] \times E\left[\phi_{kt_{mn}}\middle|\boldsymbol{T}^{\neg mn};\boldsymbol{\beta}\right].$$

A Dirichlet distribution with prior $\boldsymbol{\alpha}$ is a probability distribution that each event k has been observed $\alpha_k - 1$ times. Accordingly it can be considered that after the observation of the events $\boldsymbol{x}$, the prior $\boldsymbol{\alpha}$ is updated with the posterior $\boldsymbol{\alpha} + \boldsymbol{c_x}$, where $\boldsymbol{c_x}$ is the count vector of occurrences of each event x_k, which follows a multinomial distribution $\boldsymbol{\theta}$. Thus,

$$\boldsymbol{\theta}|\boldsymbol{x} \sim \mathrm{Dir}(\boldsymbol{\alpha} + \boldsymbol{c_x}), \qquad \text{where } \boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha}).$$

Since the expectation value of the k-th element of $\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})$ is $E[\theta_k] = \frac{\alpha_k}{\sum_i \alpha_i}$, the expectation value of θ_i given a series of events $\boldsymbol{x}$ is:

$$E[\theta_k|\boldsymbol{x}] = \frac{\alpha_k + c_{x_k}}{\sum_i(\alpha_i + c_{x_i})}.$$

Therefore,

$$P(z_{mn}=k|\boldsymbol{Z}^{\neg mn},\boldsymbol{T};\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{\alpha_k + c_{mk\cdot}^{\neg mn}}{\sum_{i=1}^{K}\alpha_i + c_{m\cdot\cdot}^{\neg mn}} \cdot \frac{\beta_{t_{mn}} + c_{\cdot kt_{mn}}^{\neg mn}}{\sum_{p=1}^{V}\beta_p + c_{\cdot k\cdot}^{\neg mn}}$$

$$= (\alpha_k + c_{mk\cdot}^{\neg mn})\frac{\beta_{t_{mn}} + c_{\cdot kt_{mn}}^{\neg mn}}{\sum_{p=1}^{V}\beta_p + c_{\cdot k\cdot}^{\neg mn}}.$$

The expected values of a per-document topic distribution and a per-topic term distribution are computed as:

$$E[\theta_{mk}] = \frac{\alpha_k + c_{mk\cdot}}{\sum_{i=1}^{K}\alpha_i + c_{m\cdot\cdot}},$$

$$E[\phi_{kv}] = \frac{\beta_v + c_{\cdot kv}}{\sum_{p=1}^{V} \beta_p + c_{\cdot k\cdot}}.$$

Hereafter, the expectation values $E[\theta_{mk}]$ and $E[\phi_{kv}]$ will be simply denoted by $\hat{\theta}_{mk}$ and $\hat{\phi}_{kv}$, respectively, for conciseness.

3.3 A Bi-LDA model

A Bi-LDA [1] model is a generative topic model in which documents are generated according to per-document topic distributions and per-topic term distributions. The difference between the Bi-LDA model and the original LDA model [13] is that the documents in Bi-LDA are aligned and per-document topic distributions are shared within the source-and-target aligned document pair whilst per-topic distributions remain separate between the source and target documents.

The merit of this separation is that this model can cover the vocabulary differences between questions and answers. For example, the word ‘why’ will occur more frequently in question documents and ‘yes’ less frequently. Furthermore, taking into account n-grams, the difference between ‘will it’ in questions and ‘it will’ in answers will be caught, meaning the Bi-LDA model can process structural information. On the other hand, the separation reduces the size of corpus to half in each side of document pairs. This may cause an inaccurate inference compared with LDA due to the data sparseness problem.

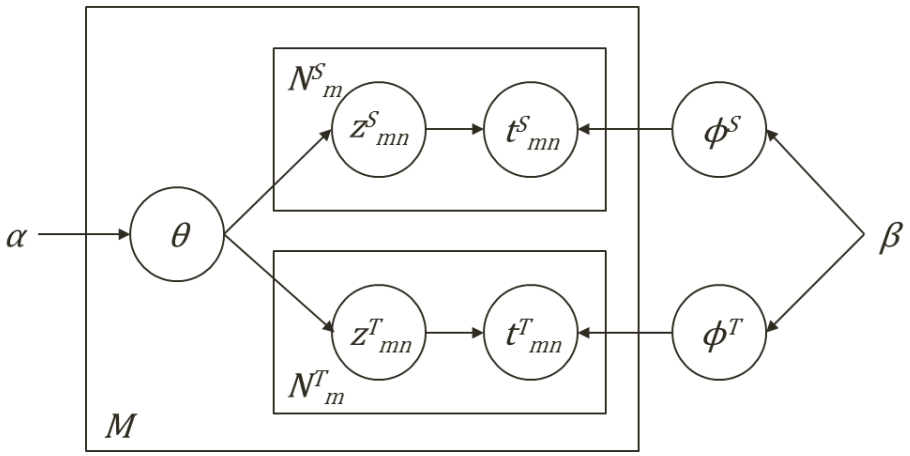

Figure 3.4. A diagram of a Bi-LDA model

The graphical model is shown in . In the m-th document pair, both the distribution of topics ${z^S}_{mn}$ and ${z^T}_{mn}$ are assumed to obey the document-pair-dependent multinomial distribution and the distribution of the term ${t^S}_{mn}$ and ${t^T}_{mn}$ are assumed to obey the source and target topic-dependent multinomial distributions, respectively. Then, both per-document topic distributions and per-topic term distributions can be generated from Dirichlet distributions, which are the conjugate priors of multinomial distributions. Thus, when the corpus includes V terms in M documents and the number of topics is set as K, K-dimensional per-document topic distributions are sampled for each document (M times) from a Dirichlet distribution with Dirichlet prior $\boldsymbol{\alpha}$ and V-dimensional per-topic term distributions are sampled for each topic (K times) from a Dirichlet distribution with the prior $\boldsymbol{\beta}$**.** In this thesis, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ will appear as scalar values because uniform Dirichlet priors are employed, while they are more generally K-dimensional and V-dimensional vectors respectively.

In this research, a Collapsed Gibbs Sampling (CGS) approach is employed, as reported by Vulić et al. [1]. Firstly, the number of topics K and Dirichlet priors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are set. Second, each term in the corpus is initialised with topics randomly. Next, a topic is sampled for each term in the corpus. For instance, in the case of the source document, for the m-th document pair, a topic ${z^S}_{mn}$ is sampled for the corresponding term ${t^S}_{mn}$ according to both the topic distribution of the m-th document pair, $\widehat{\boldsymbol{\theta}}_m$, and the term distribution of the expected source topic ${z^S}_{mn} = k$, $\widehat{\boldsymbol{\phi}}^S{}_k$, where n and k are the indexes of the term in the document and the topic, respectively. Topics in the target document are sampled via the same approach as the source document. After finishing the sampling over the entire corpus, further iterations of sampling are invoked until convergence. The summary of the procedure is as follows.

1. Set Dirichlet hyperparameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}^S$ and $\boldsymbol{\beta}^T$
2. Initialise each term with a topic
3. For each document m $(1 \leq m \leq M)$
    i. For each source term ${t^S}_{mn}$ $(1 \leq n \leq {N^S}_m)$
        - Sample the topic ${z^S}_{mn}$ for ${t^S}_{mn}$
    ii. For each target term ${t^T}_{mn}$ $(1 \leq n \leq {N^T}_m)$
        - Sample the topic ${z^T}_{mn}$ for ${t^T}_{mn}$
4. Return to 3 until convergence

In the CGS approach, sampling is done via the following formulae for uniform $\boldsymbol{\alpha}$, $\boldsymbol{\beta}^S$ and $\boldsymbol{\beta}^T$, namely $\alpha_k = \alpha$ for all k and ${\beta^S}_v = {\beta^T}_v = \beta$ for all v.

$$z^S{}_{mn} \sim \left(\alpha + c^{S\neg mn}_{mk\cdot} + c^{T\neg mn}_{mk\cdot}\right)\frac{\beta + c^{S\neg mn}_{\cdot kt_{mn}}}{V^S\beta + c^{S\neg mn}_{\cdot k\cdot}},$$

$$z^T{}_{mn} \sim \left(\alpha + c^{T\neg mn}_{mk\cdot} + c^{T\neg mn}_{mk\cdot}\right)\frac{\beta + c^{T\neg mn}_{\cdot kt_{mn}}}{V^T\beta + c^{T\neg mn}_{\cdot k\cdot}},$$

where, ${c^S}_{mkv}$ and ${c^T}_{mkv}$ follow the same rule for source and target documents, respectively, and V^S and V^T are the number of the kind of terms in the source and target vocabulary, respectively. These formulae can be derived via the same approach as those in the LDA model with the additional concept that both source and target documents are taken into account for per-document topic distributions at the same time. The first factor of each formula is proportional to the expected per-document topic distribution and the second factor corresponds to the expected per-topic term distribution (excluding the term for which a topic is currently being sampled).

Per-document topic distributions and per-topic term distributions are computed via the following formulae from the knowledge of the Dirichlet distribution:

$$E[\theta_{mk}] = \frac{\alpha + {c^S}_{mk\cdot} + {c^T}_{mk\cdot}}{K\alpha + {c^S}_{m\cdot\cdot} + {c^T}_{m\cdot\cdot}},$$

$$E\left[\phi^S{}_{kv}\right] = \frac{\beta + c^S{}_{\cdot kv}}{V^S\beta + c^S{}_{\cdot k\cdot}},$$

$$E\left[\phi^T{}_{kv}\right] = \frac{\beta + c^T{}_{\cdot kv}}{V^T\beta + c^T{}_{\cdot k\cdot}}.$$

4 Similarity Metrics

It is essential for almost all NLP systems to measure the similarity of a pair of terms or contexts since statistical machine learning approaches usually require this to match a new feature with the knowledge acquired from a training dataset. This chapter gives the basic approach to compute the similarity. Note that this research is based on the second of the following metrics (pointwise mutual information; 4.2) as subsequently detailed in the section 5.1.

4.1 Cosine Similarity

Cosine similarity is commonly used in order to estimate the similarity of two features. This is defined as the cosine of two feature vectors:

$$\mathrm{Sim}_{\cos}(\boldsymbol{a}, \boldsymbol{b}) = \cos(\mathrm{angle}(\boldsymbol{a}, \boldsymbol{b})) = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{|\boldsymbol{a}||\boldsymbol{b}|} = \frac{\sum_i a_i b_i}{\sqrt{\sum_i {a_i}^2 \sum_i {b_i}^2}}.$$

For example, Baeza-Yates and Ribeiro-Neto [88] compute the similarity of two documents via cosine metrics, where a document is represented as a vector of TF-IDF scores for each term in the document. The feature vectors can also be built using the per-topic term distribution $\boldsymbol{\phi}$ as in the last chapter. Summing over topics, the similarity for a topic model can be defined as:

$$\mathrm{Sim}_{\cos,\mathrm{topic}}\left(\widehat{\boldsymbol{\phi}}^S, \widehat{\boldsymbol{\phi}}^T\right) = \frac{1}{K}\sum_{k=1}^{K} \mathrm{Sim}_{\cos}(\widehat{\boldsymbol{\phi}}^S{}_k, \widehat{\boldsymbol{\phi}}^T{}_k).$$

4.2 Pointwise Mutual Information

A pointwise mutual information (PMI) score quantifies the degree of co-occurrence between two stochastic variables. PMI is mathematically defined as:

$$\mathrm{PMI}(t^S; t^T) = \log\frac{P(t^S, t^T)}{P(t^S)P(t^T)} = \log\frac{P(t^T|t^S)}{P(t^T)} = \log\frac{P(t^S|t^T)}{P(t^S)},$$

where t^S and t^S are terms in the case of a language model. A PMI score between a document pair is computed as the summation of all pairs of terms occurring in the document pair, namely

$$\text{PMI}(d^S; d^T) = \log\frac{P(d^T|d^S)}{P(d^T)} = \sum_{t^S \in d^S} \sum_{t^T \in d^T} \text{PMI}(t^S; t^T)$$

under the assumption that each term in a document pair is independent of the other terms in the document pair. This PMI score is called 'document correlation' in this thesis. This is introduced based on the formulation:

$$\begin{aligned}\frac{P(d^T|d^S)}{P(d^T)} &= \frac{\prod_{t^T \in d^T} P(t^T|d^S)}{\prod_{t^T \in d^T} P(t^T)} = \frac{\prod_{t^T \in d^T} P(d^S|t^T)}{\prod_{t^T \in d^T} P(d^S)} \\ &= \frac{\prod_{t^T \in d^T} \prod_{t^S \in d^S} P(t^S|t^T)}{\prod_{t^T \in d^T} \prod_{t^S \in d^S} P(t^S)} \\ &= \prod_{t^S \in d^S} \prod_{t^T \in d^T} \frac{P(t^T|t^S)}{P(t^T)}.\end{aligned}$$

In order to compute the similarity between two documents, cosine similarity can be also utilised by building the feature vectors of the PMI score for each term.

4.3 Kullback-Liebler Divergence

Kullback-Liebler (KL) divergence measures the degree of overlap between two probability distributions. This quantity is similar to distance, and accordingly it is sometimes called 'KL distance', but it is not mathematically a distance. As it is a measure of entropy, it is sometimes called the 'relative entropy' instead. The divergence is defined as:

$$D_{KL}(P||Q) = \sum_x P(X = x) \log\frac{P(X = x)}{Q(X = x)}.$$

This metric cannot be defined for Q when $Q(x) = 0$ since $D_{KL}(P||Q)$ becomes infinity. The Jensen-Shannon (JS) divergence is an improved measure for the problem using the following formulation:

$$D_{JS}(P||Q) = \frac{1}{2}\big(D_{KL}(P||R) + D_{KL}(Q||R)\big),$$

where $R(x) = (P(x) + Q(x))/2$, which will never equal 0. JS divergence has another advantage over KL divergence in that it is a symmetric. Celikyilmaz, Hakkani-Tur and Tur applied the JS divergence in order to compute the similarity of a document pair over topics [81]. They calculated the similarity as:

$$\mathrm{Sim}^{1}_{\mathrm{JS,topic}}(\boldsymbol{d}^S, \boldsymbol{d}^T) = \frac{1}{K}\sum_{k=1}^{K} W\big(\boldsymbol{\varphi}^S(\boldsymbol{d}^S, k), \boldsymbol{\varphi}^T(\boldsymbol{d}^T, k)\big),$$

where $W(P,Q) = 10^{-\delta \cdot D_{JS}(P||Q)}$ (δ is a scaling parameter e.g. $\delta = 1$) and $\boldsymbol{\varphi}(\boldsymbol{d}, k)$ is a V dimensional vector where the v-th element is $\hat{\phi}_{kv}$ if $v \in \boldsymbol{d}$, or 0 otherwise. A further definition of topic-based similarity also incorporates the per-document topic distribution $\widehat{\boldsymbol{\theta}}$:

$$\mathrm{Sim}^{2}_{\mathrm{JS,topic}}(\boldsymbol{d}^S, \boldsymbol{d}^T) = 10^{-D_{JS}(\widehat{\boldsymbol{\theta}}^S(\boldsymbol{d}^S)||\widehat{\boldsymbol{\theta}}^T(\boldsymbol{d}^T))},$$

where $\widehat{\boldsymbol{\theta}}(\boldsymbol{d})$ is the expected per-document topic distribution for the document $\boldsymbol{d}$ [81].

4.4 Kernel Tree

A kernel function can be utilised so as to compute the similarity between kernel trees as mentioned in Section 2.2.1.3. The kernel function is defined for kernel trees as:

$$K(T_1, T_2) = \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} C(n_1, n_2),$$

where N_1 and N_2 are sets of nodes in the kernel trees T_1 and T_2, respectively, and $C(n_1, n_2)$ is the number of common fragments rooted in nodes n_1 and n_2 [10]. $C(n_1, n_2)$ can be calculated via the following rule:

$$C(n_1, n_2) = \begin{cases} 0, & \text{if } n_1 \neq n_2 \\ 1, & \text{if } n_1 = n_2 \text{ and they are terminal nodes} \\ \lambda, & \text{if } n_1 = n_2 \text{ and they are preterminal nodes} \\ \lambda \prod_{j=1}^{nc(n_1)} \{1 + C(ch(n_1, j), ch(n_2, j))\}, & \text{otherwise} \end{cases}$$

where $nc(n)$ is the number of children of the node n, $ch(n, j)$ is the j-th child rooted in the node n, $n_1 = n_2$ denotes that the production rule of the nodes n_1 and n_2 is equal, and λ is a constant which penalise the scores of deep child nodes. Without employing λ, a pair of identical trees gets an extremely high score because the correspondence of large tree fragments gives exponentially large $C(n_1, n_2)$ depending on the depth of the fragments [36]. For the same reason, the kernel function is typically normalised [36], for example:

$$K'(T_1, T_2) = \frac{K(T_1, T_2)}{\sqrt{K(T_1, T_1)K(T_2, T_2)}}.$$

Wang et al. proposed an improved approach called 'syntactic tree matching' where verbs and nouns are treated as more important [10].

5 Bi-LDA Model for Answer Re-ranking

5.1 Correlation

In this research, "correlation" is defined with a pointwise mutual information (PMI) score or a variation thereof. PMI scores are computed to measure the overlap of terms for both the topic and the document. These are named topic-based PMI, $\text{PMI}_{\text{topic}}(t^S; t^T)$, and document-based PMI, $\text{PMI}_{\text{doc}}(t^S; t^T)$, respectively. Correlation can be defined for not only a pair of terms but also a pair of documents. The correlation between two documents is called 'document correlation' in this thesis to differentiate it from "term" correlation.

The advantages of PMI are:

1. simple to compute
2. easy to estimate the range of a PMI score depending on the size of corpus
3. reasonable to employ a threshold or an intercept because of the log scale
4. empirically able to obtain an intuitive relevant term ranking.

Moreover, PMI was found to perform better in this work than Park's model [83], which incorporated an IDF-based similarity ranking.

5.1.1 Topic-based PMI

One variation of correlation metric is topic-based PMI. This is introduced as follows:

$$P(t^T|t^S) \approx \sum_i^K \left\{ P(t^T|z^T = i) \frac{P(t^S|z^S = i)P(z^S = i)}{\sum_j^K P(t^S|z^S = j)P(z^S = j)} \right\}$$

$$= \frac{\sum_i^K \phi_{it^S}^S \phi_{it^T}^T P(z^S = i)}{\sum_i^K \phi_{it^S}^S P(z^S = i)}$$

$$\therefore \text{correlation}_{\text{topic}}(t^S, t^T) = \text{PMI}_{\text{topic}}(t^S; t^T)$$

$$= \log \frac{\sum_i^K \phi_{it^S}^S \phi_{it^T}^T P(z^S = i)}{\sum_i^K \phi_{it^S}^S P(z^S = i) \sum_i^K \phi_{it^T}^T P(z^T = i)},$$

when assuming that the translation probability $P(t^T|t^S)$ can be computed as [1]:

$$P(t^T|t^S) \approx \sum_{i}^{K} P(t^T|z^T = i)P(z^S = i|t^S).$$

Example correlation rankings using topic-based PMI are shown in Table 5.1. For topical terms like ‘color’ and ‘firefox’, the results of Table 5.1 indicate this correlation metric may rank terms intuitively. Similarly, for some non-topical terms like ‘why’, the metrics picked up ‘reason’ and ‘because’ as relevant terms which would lead us to expect topic-based PMI will correctly re-rank answer candidate documents. However, the ranking of related terms to ‘firefox’ may represent a misjudgement because, for example, ‘facebook’ and ‘twitter’ have no straightforward relationship. This will be disadvantageous but at the same time it can be expected to cover a wider range of relevant terms. In the case of non-topical terms like ‘is’, the rankings show nearly random results. Fortunately, the scores of non-topical terms tended to be smaller than those of topical terms. Furthermore, in the case of ‘firefox’, the correlation was relatively low for unsure terms, like ‘facebook’, in spite of the high “topicality” of ‘firefox’. This fact may help the system to ignore undesirable terms when collecting related terms.

When the number of loops for the inference was not enough, topic-based PMI would tend to create inferior term rankings. Table 5.2 shows that 100-iterations would not be adequate to rank relevant terms. The term ‘why’ was not judged to be strongly related to either ‘reason’ or ‘because’. Also, ‘shampoo’ and ‘shaving’ were listed for the term ‘color’ even though the term was topical. When it came to ‘firefox’, the relevance appeared subjectively to not be very different although the class of relevant terms was changed. On the whole, it can be considered that when the inference was iterated 500 times it was more accurate than when iterated 100 times only.

Table 5.1 The correlativity ranking for topic-based PMI of the inference iterated 500 times

is		why		color		firefox	
considered	5.30	reasons	8.08	blue	8.76	profile	7.36
here	5.17	reason	7.81	color	8.69	facebook	7.34
possible	5.09	why	7.31	brown	8.67	avatar	7.34
has	4.72	because	7.31	colors	8.63	twitter	7.28
hasn't	4.67	somehow	6.04	purple	8.61	instagram	7.26
must	4.53	simply	5.79	red	8.61	url	7.22
is	4.36	happening	5.77	pink	8.57	chrome	7.21
been	4.30	understand	5.71	grey	8.52	tumblr	7.18
← non-topical						topical →	

Table 5.2. The correlativity ranking for topic-based PMI of the inference iterated 100 times

is		why		color		firefox	
@	5.80	dom	5.21	hair	7.58	disk	7.10
question	4.94	omnisciently	5.05	dye	7.52	comptia	7.10
troll	4.91	primers	4.74	shampoo	7.49	puter	7.09
answer	4.88	we	4.64	blonde	7.37	cd	7.03
due	4.74	our	4.62	salon	7.37	linux	7.02
answered	4.72	people	4.55	shaving	7.35	boot	7.02
explanatory	4.55	disputes	4.52	scalp	7.34	install	7.01
response	4.47	foolish	4.51	shave	7.33	windows	7.01
← non-topical						topical →	

Table 5.3. The correlativity ranking for document-based PMI

is		why		color		firefox	
_?	0.98	shiddy	3.08	mim	6.92	full-screen	9.27
engineer's	0.98	jaegers	3.08	fsr	6.92	hist	9.27
mitzvah	0.98	lim_{x	3.08	carnivora	6.92	idm	9.11
65.2	0.98	garrosh	3.08	butterfly's	6.92	netsh	8.87
x_1	0.98	yazid	3.08	mammalia	6.92	anyhing	8.87
powerleveling	0.98	baal	3.08	dv7	6.92	conection	8.55
rhd	0.98	akhilleos	3.08	tatted	6.92	synchronize	8.55
struct	0.98	zanpakut	3.08	recived	6.92	pop-ups	8.55
← non-topical						topical →	

Table 5.4. The correlativity ranking for document-based PMI with a DF modifier

is		why		color		firefox	
vet	5.41	atheists	15.76	color	52.53	firefox	64.96
normal	5.12	blacks	15.71	dye	45.22	browser	57.83
relationship	5.07	christians	14.95	brown	43.70	chrome	55.40
her	4.99	racist	14.94	blonde	42.16	uninstall	49.22
she	4.88	jealous	14.70	colors	41.41	browsers	47.86
she's	4.75	because	14.67	blue	40.65	download	47.41
9.8	4.64	ignorant	14.53	colour	40.07	tab	46.66
sounds	4.56	blame	14.52	hair	39.69	firewall	46.09
← non-topical						topical →	

5.1.2 Document-based PMI

Another variation of correlation, document-based PMI, is defined with a smoothing parameter γ:

$$\begin{aligned}\text{correlation}_{\text{doc}}(t^S, t^T) &= \text{PMI}_{\text{doc}}(t^S; t^T) \\ &= \log \frac{P(t^T|t^S)}{P(t^T)} \\ &= \log \frac{N(t^S, t^T) + \gamma}{\sum_{v^T} N(t^S, v^T) + V^T\gamma} \frac{\sum_{v^T} N(v^T) + V^T\gamma}{N(t^T) + \gamma},\end{aligned}$$

where $N(t^S, t^T)$ is the count of co-occurrence of the term t^S in a source document with the term t^T in the corresponding target document, $N(t^T)$ is the count of occurrence of the term t^T in the target documents, and v^T is a term appearing in the target documents. This can be considered a more common and straightforward definition of PMI, compared with topic-based PMI.

For the purpose of generating the correlation ranking, this document-based PMI may not be suitable because infrequent terms tend to be lifted up to the top, as shown in Table 5.3. The ranking appears to have been chosen almost randomly. In order to take into account the reliability of information, the logarithm of document frequency (DF) modifier is employed:

$$\text{correlation}^{\text{DF}}_{\text{doc}}(t^S, t^T) = \text{PMI}_{\text{doc}}(t^S; t^T) \cdot \log \text{DF}(t^T).$$

Table 5.4 shows this modifier makes the document-based PMI ranking more intuitive since the metrics have captured the target term ‘because’ from the source term ‘why’, which clearly should be related. For topical terms, document-based PMI with a DF modifier is likely to find enough straightforwardly related terms compared with topic-based PMI. In particular, it can be seen that document-based PMI lists the straightforwardly related terms of ‘firefox’ more successfully than topic-based PMI. As mentioned, this can be an advantage and a disadvantage at the same time.

We found that topical terms tend to co-occur with a specific set of terms in the aligned corpus. Thus, “topicality” can be measured with a variance-like formulation:

$$\text{topicality}(t^S) = \sqrt{\sum\nolimits_{v^T} P(v^T|t^S)\{\text{PMI}_{\text{doc}}(t^S; v^T)\}^2},$$

$$\text{topicality}(t^T) = \sqrt{\sum\nolimits_{v^S} P(v^S|t^T)\{\text{PMI}_{\text{doc}}(v^S; t^T)\}^2}.$$

Here, document-based PMI is employed since topic-based PMI often produces a “non-topical” topic, meaning non-topical terms are judged as topical. indicates non-topical terms have lower topicality scores and topical ones have higher scores. These topicality metrics will be helpful to normalise document-based PMI because a topicality score prevents a document-based PMI ranking from regarding the terms which co-occur by chance as correlated. Correlation with topicality modifiers is defined as

$$\text{correlativity}_{\text{doc}}^{\text{topical}}(t^S, t^T) = \frac{\text{PMI}_{\text{doc}}(t^S; t^T)}{(\text{topicality}(t^S) + \delta) \cdot (\text{topicality}(t^T) + \delta)},$$

where δ is a smoothing parameter to prevent the score of untopical terms from becoming too high. shows results of applying PMI with topicality modifiers. We see that both ‘because’ and ‘reason’ are found as terms related to ‘why’, whereas the PMI with a DF modifier found only ‘because’. As illustrated by , rankings of correlation with topicality modifiers may be effective in capturing related terms especially for topical terms. A drawback of using this metric is that the scores for ‘is’ are relatively high even though the terms shown in the ranking are basically not related.

Table 5.5. The topicalities of question terms

is	why	problem	color	xbox	firefox
0.284	0.635	0.898	1.399	1.722	3.064
← non-topical			topical →		

Table 5.6. The correlation ranking for document-based PMI with topicality modifiers

is		why		color		firefox	
normal	1.26	because	1.33	color	1.58	firefox	0.73
vet	1.04	themselves	1.29	brown	1.31	brower	0.67
she's	1.03	reasons	1.29	blue	1.29	chrome	0.64
isn't	1.01	why	1.28	colors	1.27	bar	0.59
sounds	0.99	blame	1.19	dark	1.21	tab	0.59
she'll	0.98	jealous	1.18	red	1.20	flash	0.58
her	0.97	society	1.17	colour	1.18	download	0.52
relationship	0.97	men	1.16	dye	1.16	tools	0.57
← non-topical				topical →			

5.1.3 Document Correlation

A PMI score between the question and answer document pair is computed as derived in Section 4.2 such as:

$$\mathrm{PMI}(d^S; d^T) = \sum_{t^S \in d^S} \sum_{t^T \in d^T} \mathrm{PMI}(t^S; t^T).$$

In this research, the majority of terms tends to be assigned to the same topic because of the sharply peaked topic distribution due to the Dirichlet prior $\beta = 0.1$. As a result, almost all pairs of terms have a negative score unless the pair of terms has the same peak as the topic distribution. To address the problem, the following revised equation is defined:

$$\mathrm{PMI}^{\mathrm{positive}}(d^S; d^T) = \sum_{t^S \in d^S} \sum_{t^T \in d^T} \max\left(0, \mathrm{PMI}(t^S; t^T) - T_{\mathrm{intercept}}\right),$$

where $T_{\mathrm{intercept}}$ is the intercept of the correlation scores. In this equation, negative scores are ignored after subtracting $T_{\mathrm{intercept}}$ so that the negative correlations are not taken into account. These correlation scores can be utilised for answer re-ranking, as used in this

research, although we will see that these metrics perform worse than re-ranking only with Okapi/BM25 [26].

5.1.4 IDF-based Similarity

In order to clarify the performance of PMI, IDF-based similarity metrics are incorporated. As derived in Section 5.1.1, these $P(t^T|t^S)$ can be computed as follows:

$$P(t^T|t^S) \approx \frac{\sum_i^K \phi_{it^S}^S \phi_{it^T}^T P(z^S = i)}{\sum_i^K \phi_{it^S}^S P(z^S = i)}.$$

Then, IDF-based similarity is:

$$\mathrm{Sim}_{\mathrm{IDF,topic}}(t^S, t^T) = P(t^T|t^S) \times \log_2 \frac{M}{\mathrm{DF}(t^T)},$$

where M is the number of documents. This idea is the same as Park's work [83] except for the difference that they used the Okapi/BM25 based IDF score.

5.2 Query Expansion

Query expansion is an idea to improve the recall performance of an answer re-ranking system by taking into account further terms that are relevant to each query term as well as the query term itself. The set of expanded terms from the term t with the expansion size l, expressed as $E_l(t)$, are chosen according to one of the correlation metrics (or the IDF-based similarity as a reference). An existing approach is performed with the following formulae:

$$S(q,d) = \lambda S_E(q,d) + (1-\lambda) S_{\mathrm{BM25}}(q,d),$$

$$S_E(q,d) = \sum_{t^S \in q} \sum_{t^T \in E_l(t^S)} \frac{\mathrm{freq}(d,t^T)(k_1+1)}{K_d + \mathrm{freq}(d,t^T)} w_E(q,t^T),$$

$$w_E(q,t^T) = \sum_{t^S \in q} P(t^T|t^S) \log\left(\frac{N - \mathrm{DF}(t^S) + 0.5}{\mathrm{DF}(t^S) + 0.5}\right),$$

where q is a query, d is a document, λ is the weighting parameter for the query expansion and $S_{\mathrm{BM25}}(q,d)$ is the Okapi/BM25 score of the query q against the document d. This

model is built according to the same idea as Okapi/BM25. The Okapi/BM25 scores are accumulated for all expanded terms of the query using the weights of the translation probabilities of the original terms to the expanded terms.

Note that the results of this approach were reported by Park, et al. in 2009 [83] but they used a different approach to choose the expanded terms. They expanded a query according to the term rankings arising from the product of the translation probability $P(t^T|t^S)$ and the inverse document frequency (IDF) score, whereas in this research correlation scores are used to build rankings. In addition, Vulić et al. [1] simply incorporated the translation probability as the similarity between the source and target terms. Using PMI in order to select similar terms for query expansion is a unique difference to these studies. When using the translation probability, frequent terms tend to be highly ranked as compared to using PMI. Furthermore, when using the IDF modifier, the highest score will vary significantly depending on the term for which the ranking is built. For example, the highest topic-based PMI for each term is generally around 9.0 in this experiment as shown in Table 5.1 while the highest score in Park's model is unpredictable. This will result in difficulty in determining a threshold to expand a query.

In this research, three approaches with logarithm weights are proposed. The first approach, called 'BM25 weighted with correlation' approach, is performed with the following formulae:

$$S(q,d) = S_{\mathrm{BM25}}(q,d) + cS_E(q,d),$$
$$S_E(q,d) = \sum_{t^S \in q} \sum_{t^T \in E(t^S)} s_E(t^S, t^T, d),$$
$$s_E(t^S, t^T, d) = \max(0, \mathrm{correlation}(t^S, t^T) - T) \times S_{\mathrm{BM25}}(t^T, d),$$

where $\mathrm{correlation}(t^S, t^T)$ is one of the correlation metrics including PMI, defined in 5.1, and T is the intercept of the correlation score to help ignore terms which are not strongly correlated. Owing to the fact that the correlation scores are rescaled with a logarithm, all expanded terms are more evenly taken into account as compared to the case of using linearly scaled metrics. For instance, a PMI score 8.0 is only twice as high as 4.0 but in a linear scale the difference is 16 times. In the second approach, called the 'top- h BM25 weighted with correlation' approach, after calculating $s_E(t^S, t^T, d)$ scores for all

$t^T \in E_l(t^S)$, only the top- h highest scores are used in the formula for $S_E(q,d)$. This is given by the formula:

$$S_E(q,d) = \sum_{t^S \in q} \sum [h - \text{highest}\{s_E(t^S, t^T, d) | t^T \in E(t^S)\}].$$

This operation is expected to reduce noise due to irrelevant terms in the expanded term list. The third approach, called 'top- h BM25 with topicality-based weighting' approach, divides the problem into the case of topical terms non-topical terms based on the topicality metrics proposed in this research. Criteria such as a threshold $T_{\text{topical}} = 1.0$, are applied to separate topical and non-topical terms. Topical terms are then weighted with topic-based correlation such as $\text{PMI}_{\text{topic}}$, whilst non-topical terms are weighted with document-based correlation such as the correlation with topicality modifiers. The reason behind this use of different correlation metrics is that topic-based correlation tends to find irrelevant terms for non-topical inputs, whereas document-based correlation usually finds related terms.

5.3 Experiment

5.3.1 Experimental Settings

In the experiments of this chapter, a term was taken as a unigram tokenised by the following rules: (1) documents were separated by all blank characters such as space and newline; (2) a punctuation mark at the head or tail of a particle were separated as a new particle: e.g. '&#FF;' was parsed into the three particles '&#', 'FF' and ';'; (3) uppercase characters were converted into lowercase; and (4) series of the same punctuation marks were united so that, for example, '!!!??' became '!?'. Note that this rule sometimes does not work properly. For instance, the word 'etc.' generates an undesirable period as a term and also the word 'c#' becomes a pair of meaningless words 'c' and '#'. However, the rules allow tokenisation of the documents.

An aligned corpus for Bi-LDA inference was extracted from the SQA forum, Yahoo! Answers. A question document consisted of the subject and content of the question and the corresponding answer document was taken as the 'best answer' of the question which had been chosen by a human user. 273,397 question and answer pairs were collected from all

categories in the forum. Terms which occurred less than 10 times in the corpus were excluded. Eventually the corpus sizes were 22,503,409 question terms and 20,414,166 answer terms. The unique vocabulary sizes were 41,095 and 40,310, respectively.

To clarify the contribution of using bilingual LDA, a simple LDA model was also evaluated. The number of documents was the same while the corpus size and the unique vocabulary size were 42,917,575 and 49,985, respectively.

For Bi-LDA, the Dirichlet prior for per-document topic distributions $\boldsymbol{\theta}$ was $\alpha = 0.5$, the Dirichlet prior for both source and target per-topic term distributions $\boldsymbol{\phi}^S$ and $\boldsymbol{\phi}^T$ were $\beta = 0.1$, the number of topics was $K = 500$ and the number of iterations was either $i = 100$ or $i = 500$. Note that the number of iterations may not be adequate for convergence, but these figures were chosen because Bi-LDA is too slow to run the inference 10,000 times, for example. The correlation scores are measured using $\gamma = 0.1$ and $\delta = 0.1$. Recall these are the smoothing parameters for document-based PMI and correlation with topicality modifiers, respectively. The parameters for Okapi/BM25 were $k_1 = 0.1$ and $b = 0.75$, as reported in Isozaki's paper [27].

5.3.2 Evaluation

The answer candidates are re-ranked with 'document correlation' and 'query expansion' approaches. 216 why-questions and the corresponding 206 answer documents, aligned by Verberne in 2007 [89], were prepared for the evaluation. The questions were randomly collected from Webclopedia and the answers were collected from Wikipedia. The answer documents were re-ranked for each question and the ranks were evaluated in terms of geometric mean rank, top-1 precision, top-10 precision and mean reciprocal rank (MRR). The answer documents were also re-ranked with Okapi/BM25 without query expansion as a baseline. Note that this is a relatively easy task due to the guarantee that the answers appear in a small set of documents, namely only 206 documents.

An answer document consisted of the contents of <P>, <LI>, <H1> tags in a Wikipedia article. Each of these was called a passage. For evaluation based on 'document correlation', each passage was scored with the document correlation metrics derived in Section 4.2. Taking at most the top three scores, the average score was used as the document score. As a result, correlation scores of passages in the answer document that were irrelevant to the

question would be largely ignored. This was an important idea since such passages might have large negative scores, which could hide relevant passages. When it comes to query expansion approaches, the partitions of passages were collapsed to examine the whole document.

5.4 Result and Discussion

5.4.1 Ability to Find Similar Terms

The similarity ranking between questions and answers has been already discussed in Section 5.1. To summarise, topic-based PMI can build relatively high quality similarity rankings compared to document-based PMI. However, document-based PMI can also build good rankings by employing modifiers like document frequency (DF) and topicality.

5.4.2 Document Correlativity Scoring

Answer documents were re-ranked with document correlation metrics to give the results of Table 5.7. Overall, the performances of document correlation were considerably poorer than the baseline model, Okapi/BM25. MRR@150 was only 0.0658 for the topic-based document correlation metrics whereas it was 0.5453 for the no-expansion model. It could therefore be considered that the PMI-based model is unable to differentiate the keywords from a query while the Okapi/BM25 model could detect them thanks to the IDF modifier.

In the case of topic-based PMI, a negative correlation appears to disturb a preferable re-ranking because the geometric mean was improved from 50.91 to 39.60 by employing a positive PMI only strategy. (Although MRR@150 was not improved, this would be because of the accidental change of Success@1 which although in absolute terms was small, in relative terms was large.) It seemed that Bi-LDA with Dirichlet prior $\beta = 0.1$ tends to strongly assign each term to a specific topic. For example, the term ‘windows’ and ‘android’ should share their topics intuitively as both are operating systems but different topics were assigned by the inference in this research. The term ‘windows’ was mainly assigned into the topic #144 and the term ‘android’ was mostly assigned into the topic #290 whereas ‘windows’ and ‘android’ had never been assigned into the topics #290 and #144, respectively. Accordingly, by employing only positive scores in calculating the

document correlation, the geometric mean rank was improved. Note that the performance via topic-based PMI in Table 5.7 was obtained by using the inference iterated 100 times ($i = 100$) only. The inference iterated 500 times ($i = 500$) was also evaluated but there were differences between them as shown in Figure 5.1, Figure 5.2 and Figure 5.3.

Document-based PMI showed a better result than topic-based PMI since the co-occurrence in terms of documents would be more naturally distributed than co-occurrence in terms of topics. The DF modifier might hide the keywords, which are typically infrequent, and deteriorate the performance. The DF modifier would be better for making a related term ranking but worse for re-ranking answers.

When iterated both 500 times and 100 times, the inference with $T_{\mathrm{intercept}} = 5.0$ performed the best as shown in Figure 5.1, Figure 5.2 and Figure 5.3. This result illustrated that lower topic-based PMI scores than 5.0 were too noisy for the document correlation approach. On the other hand, at $T_{\mathrm{intercept}} = 8.0$ the accuracy of the system dropped sharply. There were very few terms which have higher topic-based PMI scores than this criteria, and consequently the system might be unable to match even the topic of the answer document due to the lack of clues. Also, because of the small expansion, unsuitable terms for the answer might dominate and deteriorate the validity of the document correlation score. Comparing the inference iterated 500 times with the one iterated 100 times on the topic-based PMI model, the former showed a better re-ranking if the intercept $T_{\mathrm{intercept}}$ was small while the latter showed a better otherwise. However, a large difference was not observed.

Table 5.7. The performance with topic-based PMI ($i = 100$) and document-based PMI

	Geo. mean	Success@1	Success@10	MRR@150
Topic-based PMI	50.91	2.31%	12.96%	0.0720
Topic-based PMI (positive)	39.60	0.93%	19.91%	0.0658
Doc-based PMI	11.63	22.69%	48.15%	0.3197
Doc-based PMI (positive)	23.82	3.70%	45.37%	0.1203
Doc-based PMI with DF	13.30	19.91%	45.37%	0.2941
Okapi/BM25	3.03	37.04%	83.80%	0.5453

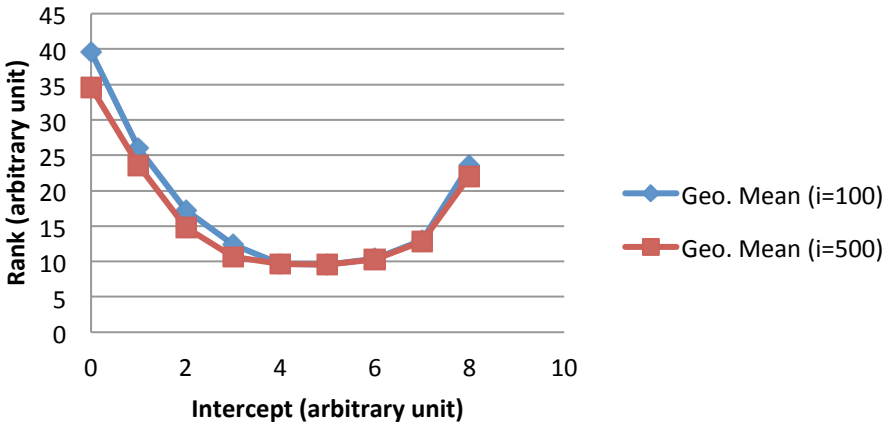

Figure 5.1. Geometric mean rank for various intercepts with topic-based PMI

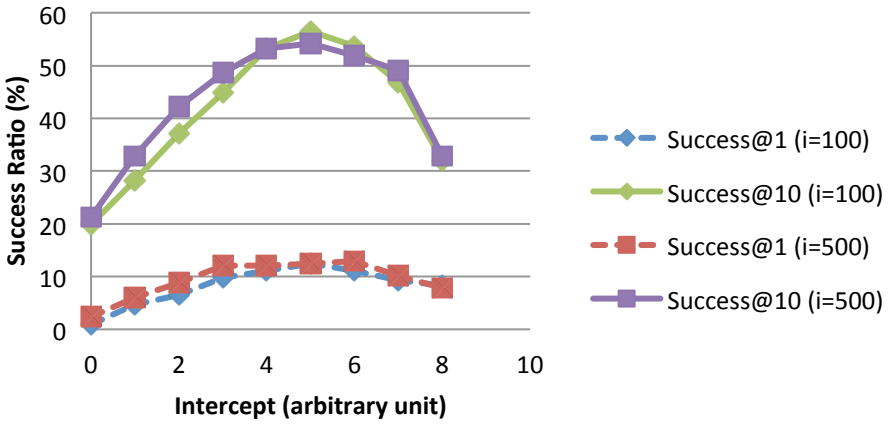

Figure 5.2. Success ratio for various intercepts with topic-based PMI

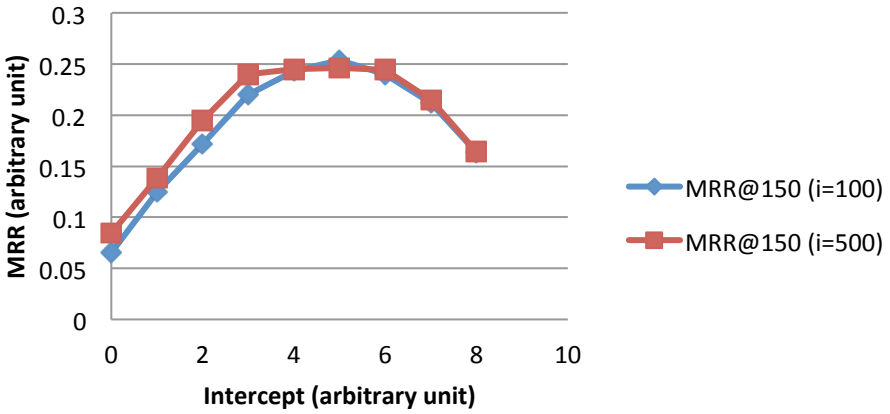

Figure 5.3. MRR@150 for various intercepts with topic-based PMI

Finally, the topic-based PMI model and a topic-based IDF-based similarity model were compared in the document correlation scoring approach. Both models performed similarly in spite of the difference of the metrics. IDF has the advantage in differentiating keywords but the translation probability $P(t^T|t^S)$ might be exaggerated for frequent terms such as 'is'. Consequently, the balance of such advantages and disadvantages results in a similar performance.

Table 5.8. The performance with topic-based PMI and IDF-based similarity ($i = 500$)

	Geo. mean	Success@1	Success@10	MRR@150
Topic-based PMI	52.07	0.93%	13.43%	0.0590
IDF-based	50.49	1.39%	11.11%	0.0570

Table 5.9. The performance of Park's baseline approach (*i*=100)

	Geo. mean	Success@1	Success@10	MRR@150
No expansion (*l*=0)	3.033	37.0%	83.8%	0.545
l=10, λ=0.05	3.009	37.5%	84.3%	0.548
l=10, λ=0.1	3.004	37.5%	83.8%	0.548
l=100, λ=0.05	3.006	37.5%	84.3%	0.548
l=100, λ=0.1	3.004	37.0%	83.8%	0.546

Table 5.10. The performance of the 'BM25 weighted with correlation' approach (*i*=100)

	Geo. mean	Success@1	Success@10	MRR@150
No expansion (*l*=0)	3.033	37.04%	83.80%	0.5453
Baseline (*l*=100, λ=0.05)	3.006	37.50%	84.26%	0.5482
T=7.0, *l*=1000, *c*=0.05	2.755	38.43%	87.04%	0.5633
T=8.0, *l*=1000, *c*=0.2	2.807	37.50%	86.57%	0.5592

Table 5.11. The performance of the 'top- h BM25 weighted with correlation' approach (*i*=100)

	Geo. mean	Success@1	Success@10	MRR@150
No expansion (*l*=0)	3.033	37.04%	83.80%	0.5453
Baseline (*l*=100, λ=0.05)	3.006	37.50%	84.26%	0.5482
T=7.0, *l*=1000, *c*=0.4, *h*=1	2.751	39.81%	86.11%	0.5687
T=6.0, *l*=1000, *c*=0.2, *h*=1	2.734	39.81%	86.11%	0.5713
T=6.0, *l*=1000, *c*=0.1, *h*=2	2.673	41.20%	87.04%	0.5806
T=6.0, *l*=1000, *c*=0.1, *h*=3	2.668	39.35%	87.96%	0.5741

Table 5.12. The performance of the 'top- h BM25 with topicality-based weighting' approach (i=100)

	Geo. mean	Success@1	Success@10	MRR@150
No expansion (l=0)	3.033	37.04%	83.80%	0.5453
Baseline (l=100, λ=0.05)	3.006	37.50%	84.26%	0.5482
T=6.0, c=0.1, h=2 (*approach 2*)	2.673	41.20%	87.04%	0.5806
T_t=6.0, c_t=0.2, T_u=1.0, c_u=0.2, h=2	2.548	43.06%	88.89%	0.5949
T_t=6.0, c_t=0.2, c_u=0.0, h=2	2.578	42.13%	88.43%	0.5897
T_u=0.5, c_u=0.5, h=1, only document-based correlation	2.954	43.06%	83.33%	0.5718

5.4.3 Query Expansion

The result of the baseline approach is shown in Table 5.9. Performance was improved by query expansion with the baseline approach over the no-expansion model, although the approach is still not particularly effective. The marginal improvement might be caused by differences in the experimental settings.

When applying query expansion with the first proposed approach 'BM25 weighted with correlation', the performance improved as shown in Table 5.10. This result can be thought to indicate that by incorporating logarithm scale metrics a wider range of similar terms could be taken into account because the difference of two numbers in the logarithm scale was smaller than that in the linear scale. Note that the number of the expanded terms whose correlation scores were $T \geq 7.0$ was expected to be less than $l = 1000$.

Table 5.11 indicates that the second approach 'top- h BM25 weighted with correlation' obtained much better scores than any of the previous approaches. In the case of $h = 1$ and $h = 2$, top-1 precision tended to be higher while in the case of $h = 3$, top-10 precision tended to be significantly higher. In other words, smaller h increased the precision and larger h increased the recall. On the other hand, the expanded query became noisy if h was too large, and led to an undesirable result. This result illustrates how a strategy of using only the top- h scores can remove unimportant expanded terms.

The third approach 'top- h BM25 with topicality-based weighting' performed the best amongst the baseline and the proposed approaches, as shown in Table 5.12. This result is evidence that non-topical terms should not be taken into account in the topic-based correlation model because by ignoring these terms the performance was better than in the

case of the second approach. Interestingly, evaluating only with document-based correlation metrics (i.e. correlation with topicality modifiers), we simultaneously obtained the highest top-1 precision and the lowest top-10 precision in Table 5.12. This result shows that the topic-based correlation has improved the coverage of related terms so that the recall of the system can increase. This also represents a possibility for further improvement by optimising how to mix the document-based and topic-based models.

We also examined a simple LDA model with the second proposed approach, i.e. 'top-h BM25 weighted with correlation'. compares the performance of the Bi-LDA model with that of a LDA model. Overall, the Bi-LDA model performed slightly more precisely than the LDA model when the number of iterations is $i = 500$. Bi-LDA has the advantage that an expanded query is expected to consist of the terms which are likely to occur in answer documents whereas it also has the disadvantage that the size of the corpus is smaller since question and answer documents are separated. The advantages and disadvantages might be almost cancelled out, but the former perhaps slightly exceeds the latter. In the case of $i = 100$, Bi-LDA obtains higher top-10 precision whereas LDA obtains higher top-1 precision. This might be triggered by the fact that Bi-LDA at least chose terms from answer documents and accordingly the probability of a serious error may be lower. Conversely LDA might be able to do the inference more precisely due to the larger corpus. Increasing the iteration count, the influence of these factors would be smaller.

Table 5.13. The comparison of the LDA and Bi-LDA models (*i*=100) for query expansion

	Geo. mean	Success@1	Success@10	MRR@150
T=7.0, *c*=1.0, *h*=1, LDA	3.371	34.72%	81.02%	0.5092
T=7.0, *c*=1.0, *h*=1, Bi-LDA	3.038	39.35%	82.87%	0.5456
T=7.0, *c*=0.4, *h*=1, LDA	2.729	42.13%	85.65%	0.5793
T=7.0, *c*=0.4, *h*=1, Bi-LDA	2.751	39.81%	86.11%	0.5687
T=6.0, *c*=0.1, *h*=2, LDA	2.722	41.67%	85.19%	0.5778
T=6.0, *c*=0.1, *h*=2, Bi-LDA	2.668	39.35%	87.96%	0.5741

Table 5.14. The comparison of topic-based and document-based PMI (*i*=500, *T*=7.0, *l*=1000)

	Geo. mean	Success@1	Success@10	MRR@150
Topic-based, *c*=0.2, *h*=3	2.681	41.67%	84.72%	0.5801
Document-based, *c*=0.3, *h*=3	2.533	46.76%	85.65%	0.6116

An important comparison of two query expansion models, based on topic-based PMI and document-based PMI, is seen in . Surprisingly, the query expansion model via document-based PMI performed better than that via topic-based PMI. This is despite the rankings from document-based PMI not being as intuitively meaningful. Indeed, the unexpected terms in the rankings seldom occurred in the question and answer documents due to the infrequency. Therefore, frequent terms played a significant role even if unsuitable infrequent terms existed in the expansion. It should be noted that a query expansion approach via document-based PMI showed lower accuracy than one via topic-based PMI in next chapter even though the experimental settings were almost the same aside from the removal of the infrequent question and answer terms which occurred less than 50 times in the set of question documents and answer documents, respectively. It is important to realise that the result of this experiment does not mean document-based PMI will always be superior to topic-based PMI. It can be assumed that which approach performs better will depend on the corpus size as well as the criteria of the minimum frequency.

Finally, the topic-based PMI metrics are compared with the IDF-based similarity with the 'BM25 weighted with correlation' approach in Table 5.15. We see firstly that Park's model with related term rankings based on the IDF-based similarity could not improve the no-expansion model. This justifies building a related term ranking with PMI. Next, the IDF-based similarity also could not get any improvement for the 'BM25 weighted with correlativity' approach. Since topic-based PMI clearly enhanced the performance from the no-expansion model, the benefit of PMI was verified.

5.5 Further Discussion

Ultimately, this experiment showed that document-based PMI, which does not require the Bi-LDA model, outperformed the topic-based PMI for both document correlation scoring and query expansion approaches. The topic model could certainly build a reasonable relevant term ranking, but this does not lead the query expansion model to perform better.

Table 5.15. IDF-based similarity with the 'BM25 weighted with correlation' approach

	Geo. mean	Success@1	Success@10	MRR@150
No expansion (l=0)	3.530	29.17%	81.48%	0.4842
IDF-based baseline (l=10, λ=0.05)	3.532	29.17%	81.48%	0.4841
IDF-based (T=0.0, l=10, c=0.02)	3.517	29.17%	81.48%	0.4831
PMI (T=7.0, l=10, c=0.05)	3.186	31.94%	84.26%	0.5115

The problem may originate from PMI. First of all, there is the fact that the document correlation scoring model did not work well while Okapi/BM25 worked somewhat better. Employing an IDF modifier in a proper way may improve the result, although here the IDF-based similarity failed to improve the system. Some measures which highlight the keywords out of a query are required to enhance the system.

The combination approach of the document and topic models achieved the best results among the approaches tested here in conjunction with the Bi-LDA model. However, this approach would still be too simple to really maximise the performance. Currently it divides the problem into topical terms or non-topical terms, but the performance might be improved by linearly combining the topical and non-topical cases.

5.6 Conclusion

This experiment revealed that a bilingual latent Dirichlet allocation (Bi-LDA) model with the corpus extracted from a social question answering (SQA) forum could improve the performance of an answer re-ranking system. On the one hand, the answer re-ranking system based on the pointwise mutual information (PMI) score between a question and answer candidates showed a poor performance compared to the baseline system based only on Okapi/BM25. On the other hand, a query expansion approach accomplished a more desirable outcome than a no-expansion approach. The three findings of these experiments that are required to improve an answer re-ranking system by query expansion are:

1. to employ the logarithm scale for the weights for query expansion,
2. to discard expanded terms except for the terms which have higher scores than the others,
3. to divide the problem into cases of topical terms and non-topical terms.

The first point encourages all terms expanded to be treated as important, the second one regards the terms which have low scores as unimportant expanded terms, and the last one addresses the disadvantage of a topic-based model, in which an appropriate topic cannot be assigned to non-untopical terms. These findings allowed the answer re-ranking system to achieve 0.595 on top-150 mean reciprocal rank (MRR@150) and 2.55 on geometric mean rank whereas a system based on Okapi/BM25 achieved 0.545 and 3.03, respectively.

6 Bi-LDA Model with N-Gram

Using N-gram has advantages against unigram in terms of semantics and syntax. For instance, ‘doubt’ and ‘no doubt’ have clearly opposite meanings. Using bigram, we can distinguish the difference and it can be expected that the accuracy of inference will be improved. Similarly, n-gram can take into account collocations. In particular, collocations of common verbs like ‘have’ and ‘get’ with prepositions will be useful since these words seem not to individually have a specific topic. Moreover, two unigrams ‘you’ and ‘are’ will be meaningless in a bag-of-words model while the bigram ‘are you’ allows a QA system to guess that a sentence will be a question. In some languages, like German and Japanese, an article (or rather a particle in Japanese) indicates how the noun phrase works in a clause in response to its predicate, namely as subject or object. For example, in Japanese, ‘watashi wa’ indicates that the speaker is the subject, i.e. ‘I’, whilst ‘watashi o’ indicates the speaker is the object, i.e. ‘me’, in the usual case. In these cases, n-gram can capture syntactic dependencies [20].

6.1 Building a Training Document

An LDA model allows researchers to employ n-grams as terms for the inference. However, some adaptations will be required due to the fact that an n-gram and an (n-1)-gram are likely to be assigned into the same topic. One possible approach is that for each original document, n documents are defined separately according to the length of n-grams, but this approach is problematic since n-grams with different lengths have never co-occurred.

In this research, only the longest n-grams are used as terms in a document and the shorter n-grams are discarded, provided these n-grams are components of the longer n-grams. The length of the longest n-grams is regulated by the experimental settings, for example $n \leq 3$. Also, since a minimum frequency of n-grams is set, sparse n-grams will not appear in the document. For instance, if the bigram ‘salty shortbread’ is sparser than the criteria of the minimum frequency, both ‘salty’ and ‘shortbread’ can be candidates for the longest n-gram regardless of the maximum n-gram length in the experimental settings.

6.2 Scoring Answer Documents

When scoring answer candidate documents, all n-grams should be taken into account. Here, a question is parsed into a set of n-grams with the following weights. The one is uniform weighting, that is, all the weights are 1. The other is with the weights computed as:

$$w(t) = \left(\frac{\sum_{w \in t} c_w}{n(t)} \right)^{-1}$$

where t is an n-gram term, w is a word forming the term t, c_w is the count of the terms which include the word w and $n(t)$ is the length of the n-gram t. For example, when the sentence "It got colder day by day" is parsed as 'it', 'got', 'colder', 'day', 'by', 'day', 'it got', 'got colder', 'colder day', 'day by', 'by day' and 'day by day', the counts of the constituent words are $c_{\text{it}} = 2$, $c_{\text{got}} = 3$, $c_{\text{colder}} = 3$, $c_{\text{day}_1} = 4$, $c_{\text{by}} = 3$ and $c_{\text{day}_2} = 3$. Then, for example, the weight for 'colder day' is

$$w(\text{colder day}) = \left(\frac{c_{\text{colder}} + c_{\text{day}_1}}{n(\text{colder day})} \right)^{-1} = \left(\frac{3+4}{2} \right)^{-1} = 0.2857 \ldots$$

The terms including 'it', i.e. 'it' and 'it got', have higher weights than this, $w(\text{it}) = 0.5$ and $w(\text{it got}) = 0.4$, respectively, because only two terms are added when scoring, whereas there are four of the terms which contain 'day_1'.

6.3 Experiment

6.3.1 Experimental Settings

A term was defined by n-grams which were collected by following the approach described in Section 6.1. The tokenisation rules were the same as the ones in Chapter 5.

An aligned corpus for Bi-LDA inference was extracted from the SQA forum, Yahoo! Answers, from all the categories. A question document consisted of the subject and content of the question and the corresponding answer document was a set of the 'best answers' of the question.

Three types of corpus were prepared for comparison with each other. The first one was a 'unigram corpus'. The question and answer document sets were individually built with unigrams which satisfied the frequency threshold of 50. For example, in the case of the question documents, all the unigrams were used as long as they appeared at least 50 times in the question documents. The second one was a 'all n-gram corpus', where all the 3-grams or shorter n-grams were employed with the frequency threshold of 50, similar to the 'unigram corpus'. The last one was the 'non-redundant n-gram corpus', which was explained in Section 6.1, with the same frequency threshold.. The n-gram corpora were weighted with the weight function unless otherwise noted, or weighted uniformly.

273,397 question and answer pairs were collected for all the corpora. For (1) 'unigram corpus', (2) 'all n-gram corpus', (3) 'non-redundant n-gram corpus', the sizes were (1) 14,445,093, (2) 33,493,678 and (3) 14,681,267 terms in the questions and (1) 12,028,231, (2) 27,152,821 and (3) 12,854,190 terms in the answers. The vocabulary sizes were (1) 11,858, (2) 86,726 and (3) 83,016; and (1) 12,699, (2) 78,894 and (3) 75,183; respectively. These figures show the number of bigrams and unigrams was larger than that of unigrams when redundancy was permitted. Also, unigrams represented only 1/8 to 1/7 of the terms among the set of unigrams, bigrams and trigrams in corpora (2) and (3). The counts of occurrence per term are important to consider. These were approximately (1) 1,218, (2) 386 and (3) 176 in the questions; and (1) 947, (2) 344 and (3) 171 in the answers, respectively. It was found that the 'non-redundant n-gram corpus' was the most sparse one among the three corpora.

For Bi-LDA, the Dirichlet prior for per-document topic distributions $\boldsymbol{\theta}$ was $\alpha = 0.5$, the Dirichlet prior for both source and target per-topic term distributions $\boldsymbol{\phi}^S$ and $\boldsymbol{\phi}^T$ was $\beta = 0.1$, the number of topics was $K = 500$ and the number of iteration was $i = 100$. The other settings for the answer re-ranking were the same as those in Chapter 5. When initialising Bi-LDA inference, a topic for each unigram was assigned randomly according to the unsmoothed per-term topic distributions (not to be confused with the per-topic term distributions) which were inferred in Chapter 5. For both bigrams and trigrams, the assignment was done with a uniform distribution. Note that the topics of all the terms in the 'unigram corpus' were sampled from the converged topic distributions whereas the topics in the other corpora were not. This may provide a strong advantage to the 'unigram corpus' model.

6.3.2 Evaluation

Evaluation proceeded the same as the experiment in Chapter 5 except that scoring was performed with the n-gram based model shown in this chapter. For this scoring, n-grams had been added to the test set as well as the training corpora. Here, infrequent terms and redundant shorter n-grams were not excluded.

6.4 Result and Discussion

6.4.1 Ability to Find Similar Terms

The relevant term rankings over the three corpora are shown in Table 6.1, Table 6.2 and Table 6.3. Basically, the tendencies seen in the results of Chapter 5 were preserved, although the terms related to ‘why’ were destroyed when inferring via the ‘non-redundant n-gram corpus’ due to the elimination of redundant shorter n-grams. While there were 38,332 instances of ‘why’ in the question documents in the ‘all n-gram corpus’, there were only 1,375 in the ‘non-redundant n-gram corpus’. This extreme difference in term frequencies has destroyed the correlation between, for example, ‘why’ and ‘reason’.

In Table 6.4, n-gram terms including ‘why’ were ranked with topic-based PMI. This table reveals that a desirable result could not be obtained for these terms. It should be noted that the inference was done with 100 iterations only and it is possible that the system would work better after more iterations.

Although it might be said that reasonable results were obtained for the other terms, an interesting result here was the ranking for ‘firefox’. The set of related terms for ‘firefox’ varied depending on the corpus. It seemed that there was no exactly suitable topic like “browser” in the inference and the term was pulled among loosely related topics.

Table 6.1. The relevant term rankings via PMI_{topic} over the 'unigram corpus'

is		why		color		firefox	
comic	3.52	reasons	7.31	blue	8.48	disk	6.86
possible	3.50	reason	6.87	color	8.44	install	6.82
hasn't	3.47	why	6.21	brown	8.41	files	6.81
superman	3.43	our	5.94	colors	8.35	uninstall	6.77
exact	3.40	we're	5.93	red	8.32	linux	6.77
'.	3.34	ourselves	5.90	purple	8.26	reinstall	6.76
batman	3.32	we've	5.65	pink	8.21	windows	6.74
',	3.27	us	5.59	bright	8.19	malware	6.67
← untopical						topical →	

Table 6.2. The relevant term rankings via PMI_{topic} over the 'all n-gram corpus'

is		why		color		firefox	
nope	4.78	reason	8.37	blue	8.81	youtube	6.81
nope .	4.56	the reason	8.35	color	8.81	videos	6.75
nope ,	4.51	. the reason	8.27	the color	8.77	on youtube	6.75
yep	4.46	no reason	8.27	colors	8.66	youtube .	6.56
is not a	4.33	a reason	8.27	dark	8.66	video .	6.56
it is not	4.30	is because	8.27	and white	8.63	the video	6.55
superman	4.27	because of the	8.27	color .	8.63	a video	6.51
is not the	4.20	reason .	8.27	color ,	8.62	video	6.44
← untopical						topical →	

Table 6.3. The relevant term rankings via PMI_{topic} over the 'non-redundant n-gram corpus'

is		why		color		firefox	
high	4.29	countries	4.57	color	8.07	file	6.23
velocity	4.22	. we	4.55	the color	8.06	files	6.22
m/s	4.16	, we	4.51	colors	8.01	the file	6.10
v =	4.15	if we	4.49	colour	7.97	adobe	6.08
the angle	4.08	a long time	4.48	purple	7.97	the video	6.00
f =	4.02	we are	4.47	color .	7.93	linux	5.94
accelaration	3.97	european	4.46	color ,	7.83	the files	5.93
t =	3.96	that we	4.46	pale	7.79	downloading	5.92
← untopical						topical →	

Table 6.4. The relevance rankings of n-grams with 'why' over the non-redu. n-gram corpus

why do (1252 times)		why does (711 times)		why did (411 times)		why have (128 times)	
women	5.83	to do with	4.90	the war	6.00	the future .	5.01
that people	5.85	nothing to do	4.85	war ,	5.82	in the future	5.00
women .	5.79	has nothing to	4.81	war	5.76	the future ,	4.56
men	5.78	do with the	4.76	war .	5.70	or may not	4.45
people are	5.75	nothing to worry	4.70	century	5.70	may or may	4.42
a woman	5.74	worry about .	4.69	hitler	5.69	may not be	4.38
women are	5.73	have nothing to	4.63	troops	5.64	you would have	3.55
men .	5.72	or may not	4.55	had been	5.62	would have to	3.49

6.4.2 Document Correlation Scoring

The document correlation scoring approaches were examined for n-gram based answer re-ranking. Overall, the unigram based model performed generally better than the n-gram based models except for the model with the 'non-redundant n-gram corpus' using topic-based PMI with no intercept, as shown in Table 6.5 and Table 6.8. It was found that the simple n-gram models were unable to improve accuracy and in fact deteriorated it. The sparseness of bigrams and trigrams is undoubtedly the reason for this, considering the tendency for accuracy to become lower and lower when the corpus became sparser and sparser. Since document correlation tends to score rare terms highly, bigrams and trigrams were likely to be treated as important terms regardless of whether or not the terms were really important. In other words, the document correlation scoring approach might choose an answer candidate document according to infrequent terms as opposed to important terms.

An important fact to consider was that the lack of some keywords from the last experiment in this experiment with the 'unigram corpus'. For instance, since the term 'sleepwalk' occurred less than 50 times in the question and answer instances, it did not appeared in the corpora in this experiment while it appeared in the corpus in Chapter 5. This seemed potentially problematic but ultimately it did not influence the result and the system worked regardless of their absence. It could be that important keywords tended to be hidden behind unimportant terms in the PMI based answer re-ranking.

In the case of topic-based PMI models, the unigram-based re-ranking recorded the best performance when running with the intercept $T_{\text{intercept}} = 5.0$ in Figure 6.1, Figure 6.2 and

Figure 6.3. The reason might be that each term occurred relatively often in 'unigram corpus' as compared to the other corpora. Also, it should be noted that in this experiment the Bi-LDA inference was iterated only 100 times, and 'unigram corpus' had a significant advantage because all the terms in 'unigram corpus' were initialised according to the topic distribution in the inference that was iterated 500 times in Chapter 5, whereas all the bigrams and trigrams were initialised randomly for the other corpora.

Table 6.5. The performance with $\mathrm{PMI}_{\mathrm{topic}}$ for each corpus

	Geo. mean	Success@1	Success@10	MRR@150
Last experiment (non-redu.)	50.91	2.31%	12.96%	0.0720
Last experiment (positive)	39.60	0.93%	19.91%	0.0658
Unigram	48.12	1.85%	17.13%	0.0744
Unigram (positive)	42.29	1.39%	16.20%	0.0624
All n-gram	44.67	5.09%	18.06%	0.0977
All n-gram (positive)	34.34	2.31%	23.15%	0.0869
Non-redundant n-gram	26.78	7.41%	31.48%	0.1459
Non-redundant (positive)	44.70	1.85%	16.67%	0.0634

Table 6.6. The detailed performance with $\mathrm{PMI}_{\mathrm{topic}}$ for the 'all n-gram corpus'

	Geo. mean	Success@1	Success@10	MRR@150
w/ weighting function	44.67	5.09%	18.06%	0.0977
w/ w. function (positive)	34.34	2.31%	23.15%	0.0869
Uniform weighting	53.88	2.78%	14.81%	0.0658
Uniform weighting (pos.)	39.95	2.31%	16.67%	0.0725
Unigram only	52.55	3.24%	15.28%	0.0708
Unigram only (positive)	42.82	1.39%	17.13%	0.0621

Table 6.7. The detailed performance with $\mathrm{PMI}_{\mathrm{topic}}$ for the 'non-redundant n-gram corpus'

	Geo. mean	Success@1	Success@10	MRR@150
w/ weighting function	26.78	7.41%	31.48%	0.1459
w/ w. function (positive)	44.70	1.85%	16.67%	0.0634
Uniform weighting	34.07	4.17%	23.15%	0.1054
Uniform weighting (pos.)	50.38	1.39%	13.89%	0.0524
Unigram only	33.51	1.39%	22.69%	0.0839
Unigram only (positive)	52.08	1.39%	14.35%	0.0495

Without an intercept and a threshold, the ‘non-redundant n-gram corpus’ showed the best performance as shown in Table 6.5. This was an irregular result as compared to the results with intercepts. Looking at the results for the ‘non-redundant n-gram corpus’ with only unigrams in Table 6.7, a phenomenon was clear: the case without intercept obtained a better result than the case with only positive PMI scores and this was true regardless of bigrams and trigrams. It therefore seemed that it was triggered by unigrams rather than bigrams and trigrams. Firstly, by removing the redundant unigrams, negative PMI scores of the non-redundant unigrams might come to penalise undesirable documents for the answer of the question. Conversely, it can be considered that the positive correlation would be less helpful in the ‘non-redundant n-gram corpus’ than in the ‘unigram corpus’ and the ‘all n-gram corpus’ since the majority of unigrams in the ‘non-redundant n-gram corpus’ were removed and consequently did not exist in the document where the unigrams should be present. Note that this explanation might contradict the case of using document-based PMI where the ‘non-redundant n-gram corpus’ performed the worst.

Comparing the ‘unigram corpus’ with the ‘all n-gram corpus’ in Figure 6.1 to Figure 6.3, the former outperformed the latter for a well-chosen intercept whereas it did not at low intercepts like $T_{\text{intercept}} = 0.0$ or high intercepts like $T_{\text{intercept}} = 8.0$. This might be a consequence of the weighting for n-grams. As an IDF modifier is commonly utilised in the field of information retrieval, important keywords tend to be infrequent. The weighting method in this experiment often gives 1.0 to infrequent terms and lower values to the frequent terms since infrequent terms do not usually make n-grams under the experimental settings and c_w and $n(t)$ become 1.0 and 1, respectively. As a result, keywords were often weighted as important and would bring a steady result across all intercept values.

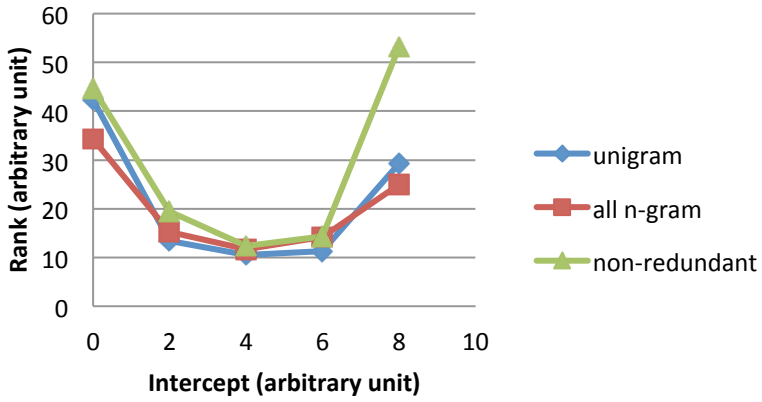

Figure 6.1. Geometric mean rank for various intercepts via topic-based PMI

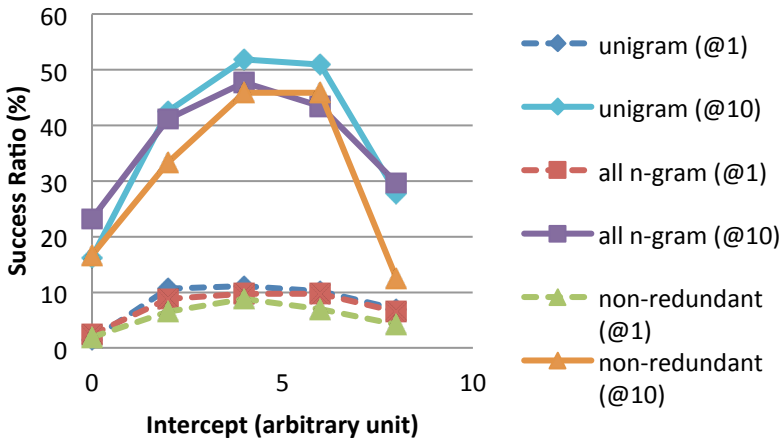

Figure 6.2. Success ratio for various intercepts via topic-based PMI

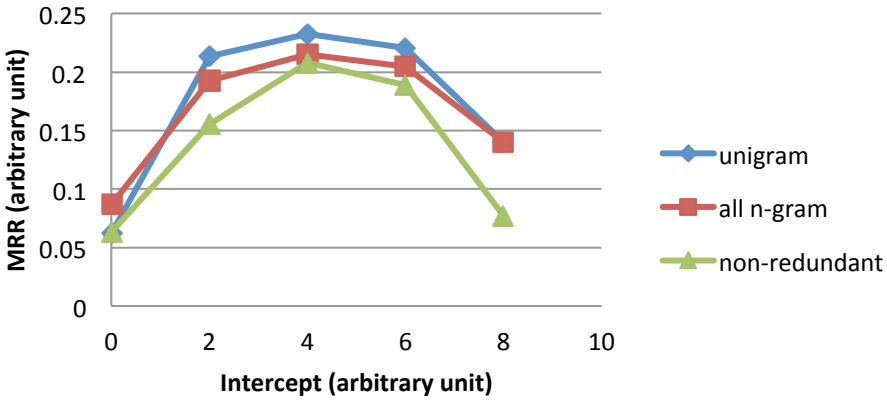

Figure 6.3. MRR@150 for various intercepts via topic-based PMI

When only positive PMI scores were employed, the ‘unigram corpus’ and the ‘non-redundant n-gram corpus’ were below the ‘all n-gram corpus’ in terms of accuracy regardless of the weighting method, as shown in Table 6.5, Table 6.6 and Table 6.7. This indicated n-grams could make some improvement, but it also indicated that non-redundancy did not help the system improve. One explanation is that the removal of redundant unigrams might hide some important unigrams and would deteriorate the quality of inference on unigrams since important correlations are weakened. In this model, unigrams can be assumed to be more helpful to find the corresponding answers to the questions since a keyword tended to be infrequent and formed a unigram instead of a bigram or a trigram. In other words, the removal disturbed the ability of unigrams to obtain a good result rather than letting bigrams and trigrams support the unigrams.

In the case of document-based PMI models, it also seemed that the ‘unigram corpus’ suited answer re-ranking, as shown in Table 6.8. The ‘unigram corpus’ worked better than even the corpus in Chapter 5, in which less frequent terms were employed. The improvement in reliability of the terms in the ‘unigram corpus’ might enable such a good result as the terms were guaranteed to appear 50 times in the corpus. Comparing the case where terms were weighted via the weighting function to the case where they were weighted uniformly in Table 6.9 and Table 6.10, the former showed a better result. In terms of MRR@150, which was 0.24 for the ‘all n-gram corpus’ and 0.16 for the ‘non-redundant n-gram corpus’, these were improved to 0.29 and 0.21, respectively, by employing the weighting function.

On the other hand, the ‘non-redundant n-gram corpus’ showed considerably worse accuracy even when it was compared with ‘all n-gram corpus’. Considering the fact that the ‘unigram corpus’ performed the best, it seemed that unigrams would play an essential role to re-rank answer candidate documents. In the ‘non-redundant n-gram corpus’, more than a half of unigrams were removed as redundant n-grams. As mentioned previously, it can be thought that such a massive removal ruins the accuracy of document correlation scores and consequently the performance of answer re-ranking as well.

Table 6.8. The performance with $\mathrm{PMI}_{\mathrm{doc}}$ for each corpus

	Geo. mean	Success@1	Success@10	MRR@150
Last experiment	11.63	22.69%	48.15%	0.3197
Last experiment (positive)	23.82	3.70%	45.37%	0.1203
Unigram	8.53	22.89%	57.87%	0.3511
Unigram (positive)	21.60	7.41%	31.94%	0.1537
All n-gram	13.27	18.98%	45.37%	0.2934
All n-gram (positive)	27.25	5.09%	30.09%	0.1244
Non-redundant n-gram	22.83	14.35%	33.80%	0.2135
Non-redundant (positive)	38.18	2.31%	18.98%	0.0801

Table 6.9. The detailed performance with $\mathrm{PMI}_{\mathrm{doc}}$ for the 'all n-gram corpus'

	Geo. mean	Success@1	Success@10	MRR@150
w/ weighting function	13.27	18.98%	45.37%	0.2934
w/ w. function (positive)	27.25	5.09%	30.09%	0.1244
Uniform weighting	17.20	16.20%	37.96%	0.2444
Uniform weighting (pos.)	34.52	2.31%	20.83%	0.0895
Unigram only	12.36	20.83%	47.69%	0.3012
Unigram only (positive)	18.28	9.26%	36.11%	0.1803

Table 6.10. The detailed performance with $\mathrm{PMI}_{\mathrm{doc}}$ for the 'non-redundant n-gram corpus'

	Geo. mean	Success@1	Success@10	MRR@150
w/ weighting function	22.83	14.35%	33.80%	0.2135
w/ w. function (positive)	38.18	2.31%	18.98%	0.0801
Uniform weighting	30.98	9.72%	24.54%	0.1569
Uniform weighting (pos.)	46.24	1.39%	13.43%	0.0594
Unigram only	22.67	14.35%	31.94%	0.2053
Unigram only (positive)	38.35	1.85%	18.52%	0.0749

A further point arises through looking at Success@1 to compare the case of all PMI scores with that of positive PMI scores There was more than a six-fold difference for the 'non-redundant n-gram corpus' whereas there was around a three-fold difference for the 'unigram corpus' and the 'all n-gram corpus'. The disparity might indicate that PMI scores on the 'non-redundant n-gram corpus' had lost important positive correlations because a unigram was removed and there only existed either a bigram or a trigram which included

the unigram. This means the correlation of the unigram in the document could not be taken into account, similar to the case of the document correlation scoring approach.

In conclusion, n-grams were not helpful in the experimental settings probably because of the sparseness of bigrams and trigrams. The sparseness would lead to unimportant bigrams and trigrams being regarded as important terms because document correlation tends to give higher scores to infrequent terms. Furthermore, removing the redundancy of shorter n-grams deteriorated the accuracy. One of the merits of the removal was that the set of terms which share the words composing the terms is not expected to be assigned to the same topics as compared to the case of employing all the n-grams. However, in this experiment, the negative effect of reducing the number of unigrams far outweighed this merit.

6.4.3 Query Expansion

Contrary to the results for document correlation, query expansion showed that the n-gram corpora outperformed the 'unigram corpus' as seen in Table 6.11 and Table 6.12. This cannot be attributed to the query expansion approaches themselves but to the baseline, namely the result of the no-expansion approach. Table 6.13 shows this achieved a high level for the n-gram based models in terms of the accuracy of the re-ranking. For example, the baseline of the 'all n-gram corpus' almost reached 32% in Success@1 while that of 'unigram corpus' did not even reach 18%. The other indices also showed significant differences between the 'unigram corpus' and the n-gram corpora. Although it seemed that bigrams and trigrams improved the query expansion system, this result was probably brought about by the weighting function because there was little difference between the case of employing the uniform weighting approach and the case of employing only unigrams in the n-gram corpora (Table 6.13). As previously mentioned, this would be because keywords tend to be less frequent than unnecessary terms. This mechanism can be thought to improve the performance with n-gram corpora rather than the existence of bigrams and trigrams in the corpora improving it.

Compared with the experiment in Chapter 5, the 'unigram corpus' showed far less accuracy. Although the previous experiment achieved 41.7% on Success@1 and 0.58 on MRR@150, this experiment achieved only 20.4% and 0.37, respectively, under the same parameters as the last experiment. The main reason for this may be that some of the important keywords have disappeared due to the higher minimum term frequency in the

corpora. The example of ‘sleepwalk’ was mentioned in Subsection 6.4.2. This was problematic here since ‘sleepwalk’ is an essential keyword for finding the corresponding document, which refers sleepwalking.

Table 6.11. The performance for topic-based PMI (T=7.0, l=1000, c=0.2, h=3)

	Geo. mean	Success@1	Success@10	MRR@150
Unigram	5.897	20.37%	69.44%	0.3716
All n-gram	5.540	25.46%	67.13%	0.3990
All n-gram (uniform weight)	9.378	16.20%	50.46%	0.2805
All n-gram (unigram only)	5.963	19.44%	67.13%	0.3616
Non-redundant n-gram	4.375	33.80%	75.00%	0.4801
Non-redu. (uniform weight)	6.612	17.13%	67.13%	0.3284
Non-redu. (unigram only)	6.556	18.19%	65.28%	0.3488

Table 6.12. The performance for document-based PMI (T=7.0, l=1000, c=0.3, h=3)

	Geo. mean	Success@1	Success@10	MRR@150
Unigram	6.258	19.91%	67.13%	0.3657
All n-gram	4.361	36.11%	75.00%	0.4913
All n-gram (uniform weight)	6.529	16.20%	65.74%	0.3320
All n-gram (unigram only)	6.499	18.52%	66.67%	0.3528
Non-redundant n-gram	4.686	34.26%	73.15%	0.4732
Non-redu. (uniform weight)	7.072	14.35%	63.43%	0.3108
Non-redu. (unigram only)	6.917	17.13%	64.35%	0.3375

Table 6.13. The performance of no expansion model

	Geo. mean	Success@1	Success@10	MRR@150
Unigram	6.846	17.59%	64.48%	0.3390
All n-gram	4.690	31.94%	73.61%	0.4616
All n-gram (uniform weight)	7.251	12.96%	62.96%	0.2997
All n-gram (unigram only)	6.847	17.59%	64.81%	0.3390
Non-redundant n-gram	4.861	31.48%	73.15%	0.4547
Non-redu. (uniform weight)	7.449	12.96%	62.04%	0.2960
Non-redu. (unigram only)	7.141	16.67%	63.43%	0.3288

Interestingly, for the ‘unigram corpus’, the topic-based PMI model performed better than the document-based one despite the opposite result in Chapter 5. The geometric mean rank of 5.90 for topic-based PMI became 6.26 for document-based PMI. This is also likely to be caused by the lack of keywords. The experiment showed that the effect of infrequent keywords was apparently more serious for document-based PMI than topic-based PMI. It might be therefore concluded that topic-based PMI would work adequately for frequent terms but it would not for infrequent terms. That is, by the removal of infrequent terms, the query expansion via topic-based PMI could improve the baseline system more than that via document-based PMI.

When it comes to the ‘non-redundant n-gram corpus’ with topic-based PMI, the query expansion system re-ranked answer candidate documents better than the system trained on the ‘all n-gram corpus’. This is in contrast to the case of the document correlation scoring. In terms of MMR@150, for example, the ‘non-redundant n-gram corpus’ enabled the system to obtain 0.48 whereas the ‘non-redundant n-gram corpus’ enabled it to obtain only 0.40. Recall that the intention of the removal of redundancy was to capture the difference in topics between shorter and longer n-grams due to the fact that the removal avoids the n-gram pair co-occurring. The results suggest this was successful. Evaluating the system only with unigrams, the ‘non-redundant n-gram corpus’ would not perform well compared to the ‘all n-gram corpus’. The MMR@150 was 0.35 and 0.36, respectively. This fact would indicate the improvement on the ‘non-redundant n-gram corpus’ is probably due to the non-redundancy feature. Another piece of evidence was the low performance for the ‘all n-gram corpus’ via uniform weighting. The MMR@150 was 0.28, whereas for the ‘non-redundant n-gram corpus’ it was 0.33. This illustrated that the improvement was not from the weighting function (unlike when the n-gram corpora outperformed the ‘unigram corpus’). Although in the case of document-based PMI the ‘all n-gram corpus’ was better than the ‘non-redundant n-gram corpus’, the explanation above did not contradict the result. That is, the non-redundant feature would not deteriorate the accuracy for document-based PMI since the document-based approach was not inferred via topics, which will be collapsed by redundant n-grams.

6.5 Further Discussion

In terms of the document correlation scoring model, it was reconfirmed that the weights for unnecessary terms should be revised. First of all, it is essential to improve the treatment of longer n-grams since the 'unigram corpus' based approach performed the best. A reasonable strategy is to evaluate scores for longer n-grams first then for shorter ones if the longer n-grams are not found, instead of evaluating all the range of n-grams at the same time with a weighting function. This strategy is expected to help the system to avoid redundantly scoring a constituent word of n-grams and keep important n-grams from being penalised with a weighting function.

When it comes to the query expansion approach, removing redundant terms was beneficial for topic-based PMI. In order to improve it, the strategy suggested above for the document correlation scoring approach may also be helpful. Furthermore, applying document-based PMI for infrequent terms and topic-based PMI for frequent terms will be another reasonable strategy based on the difference between the results for the corpus in Chapter 5 and the 'unigram corpus' in this chapter. An important fact is that this strategy is opposed to the combined approach of topic-based and document-based PMI for query expansion shown in Chapter 5. There, topic-based PMI was incorporated for topical terms but the terms which have a high topicality score tended to be infrequent. There is a need to reconsider this.

6.6 Conclusion

To conclude, n-gram corpora were helpful for the query expansion approach whereas they were not for the document correlation scoring approach. Specifically, Okapi/BM25 worked well for n-grams rather than the correlation for n-grams.

For the document correlation scoring approach, the correlation metrics did not suit n-grams. It can be considered that the difference in performance might come from the overestimation of long n-grams in terms of the importance due to the nature of the PMI metrics.

By removing redundant shorter n-grams we were able to revise the query expansion system although it adversely affected the document correlation based system. The improvement

might be evidence that any n-gram should be excluded which shares a constituent word with another n-gram in the document. In addition, it was found that the weighting function dramatically improves both of the approaches compared to uniform weighting when incorporating n-grams. However, there is a need for a number of devices which further improve the answer re-ranking system based on n-grams.

Another important finding here was that performance of the topic-based query expansion approach exceeded the document-based one in the case of the ‘unigram corpus’ unlike the experiment in Chapter 5. A topic-based model might perform better for frequent terms, considering that the most significant difference between the two experiments was the minimum frequency of terms. This will be a hint for constructing a combination model from a topic-based model and a document-based model.

7 Bi-LDA Model with Topic Replacement

Topics in a LDA model will express their semantic features of the terms, i.e. terms with the same topic will be semantically related. By replacing topical terms with topic identifiers, the data sparseness problem will be addressed, which is one of the challenging problems in the field of NLP. That is, infrequent terms will be substituted for frequent terms in which infrequent constituent words are replaced with the corresponding topic identifiers. For example, topic replacement will treat 'android users' and 'smartphone users' as an identical term, such as 'TOPIC#1 users' for example, where the topic identifier 1 tends to be assigned to both 'android' and 'smartphone'. In this experiment, a new corpus will be built using topic replaced documents and new topics will be assigned topic replaced terms such as 'TOPIC#1 users'. The term 'TOPIC#1 users' is more frequent than the original terms since the topic identifier 'TOPIC#1' is almost guaranteed to be more frequent than both 'android' and 'smartphone'. This fact is expected to lead to more accurate inference.

A topic replacement system runs LDA inference twice, firstly to determine topics for topic replacement and secondly to infer new topics for topic-replaced terms. Topic replacement will be done only for the topical words whose topicalities exceed a criterion. The two pieces of inference do not have to be run on the same corpus. For instance, the system can incorporate Wikipedia for the first inference and Yahoo! Answers for the second one. That means, if the main corpus is not large enough, a different larger corpus can be employed to learn the topic replacement mapping.

7.1 Building a Training Document

A document is built with the same approach as in Chapter 6 except that topic-replaced terms are added. That is, there exist both topic-replaced terms and original (unreplaced) terms in each document. It is justified to add the replaced terms to the set of original terms since this is a bag-of-words model. The topic-replaced terms are prepared by replacing all topical unigrams in the original with n-grams. It is the same as the case in the last chapter that shorter n-grams are removed if they are a part of the longer n-grams appearing in the document.

7.2 Scoring Answer Documents

Firstly a question is parsed into n-gram terms, following the approach in Chapter 6, and secondly topic-replaced terms are added according to the parsed terms. The weighting formula in Section 6.2 can be used directly here as well, although there is a need to note that a topic-replaced word w is included in the count c_w. For instance, when 'smartphone' is replaced with 'TOPIC#1' and 'smartphone' appears 3 times in the parsed terms but due to the appearance of another related term 'TOPIC#1' appears a total of 4 times, then $c_{\text{smartphone}} = 3 + 4 = 7$.

7.3 Experiment

7.3.1 Experimental Settings

The definition of a term was the same as the one in Chapter 6 except for the topic replacement. The topicality criterion for topical terms was $T_{\text{topical}} = 1.0$. The topic identifiers with which topical terms were replaced were taken as the most frequent topics in the inference iterated 500 times in chapter 3.

An aligned corpus for the Bi-LDA inference was prepared through the same procedure as the previous experiment in Chapter 6 except for the existence of topic-replaced terms in the documents. It should be noted that the minimum criterion of term frequency, namely 50 terms, was applied after the topic replacement. This allowed an infrequent term to appear in the corpus as a topic-replaced term. A unigram based corpus 'topic-replaced unigram corpus' and a trigram based corpus 'topic-replaced n-gram corpus' were built. The corpora were evaluated and compared with the 'unigram corpus' and 'non-redundant n-gram corpus' in Chapter 6. There were 273,397 question and answer pairs in the corpora which were collected from all the categories in the SQA forum. The 'topic-replaced unigram corpus' and the 'topic-replaced n-gram corpus' consist of 17,370,163 and 20,474,312 terms, respectively, in the question documents; 14,865,068 and 18,929,941 terms, respectively, in the answer documents. Also, the numbers of unique terms were 12,353 and 114,848 terms, respectively, in the question documents; and 13,190 and 121,463 terms, respectively, in the answer documents. Here, the counts of appearance per term were 1,406

and 178 times, respectively, in question documents; and 1,127 and 156 times, respectively, in answer documents.

The parameters for Bi-LDA and the initialisation policy were inherited from the last experiment. Again, note that the topics of all the unreplaced unigrams in the corpora were sampled from the converged topic distributions whereas the topics for the other kinds of terms were not. Consequently, the inference for 'topic-replaced n-gram corpus' was not converged perfectly because the inference was only iterated 100 times.

7.3.2 Evaluation

Evaluation proceeded in the same manner as the previous experiments. The difference was that topical terms in the test set had been replaced with the corresponding topic identifier in the same way as in the training corpora.

Table 7.1. The relevant term rankings via $\mathrm{PMI}_{\mathrm{topic}}$ over the 'topic-replaced unigram corpus'

is		why		color		firefox	
TOPIC#469	4.58	reasons	8.00	TOPIC#450	8.46	twitter	7.36
considered	4.58	TOPIC#452	8.00	color	8.45	profile	7.36
TOPIC#43	4.22	reason	6.90	dark	8.43	block	7.25
impossible	4.21	TOPIC#311	6.45	blue	8.42	blog	7.19
TOPIC#424	4.00	ourselves	6.35	colors	8.36	instagram	7.19
TOPIC#62	4.00	ourselves	6.30	pink	8.35	pic	7.14
gives	3.98	dang	6.29	green	8.34	ads	7.13
itself	3.98	we'll	6.25	brown	8.33	tumblr	7.11
← non-topical						topical →	

Table 7.2. The relevant term rankings via $\mathrm{PMI}_{\mathrm{topic}}$ over the 'topic-replaced n-gram corpus'

is		why		color		firefox	
considered	6.25	rules	5.57	TOPIC#450 ,	8.04	your TOPIC#148	7.34
TOPIC#469	6.21	TOPIC#194	5.56	color	8.03	TOPIC#148 TOPIC#148	7.30
. TOPIC#447 ,	6.14	. we	5.53	TOPIC#450 and	8.02	a TOPIC#148	7.23
. therefore ,	6.00	we are	5.51	the TOPIC#450	8.01	TOPIC#148 is	7.21
. otherwise ,	5.94	, we	5.47	the color	8.00	instagram	7.19
. thus ,	5.62	what we	5.45	TOPIC#450	7.98	your profile	7.18
TOPIC#447 , you	5.40	if we	5.41	TOPIC#450 TOPIC#450	7.96	twitter	7.18
otherwise , you	5.11	our	5.41	colors	7.95	TOPIC#148 ,	7.11
← non-topical						topical →	

7.4 Result and Discussion

7.4.1 Ability to Find Similar Terms

The rankings, shown in Table 7.1 and Table 7.2, were basically similar to those in the experiment in Chapter 6. It can be considered that building term rankings with topic identifiers is a reasonable approach. However, note that topic-based PMI used with the topic-replaced corpora could not capture 'because' from 'why' in the rankings of Table 7.2. No clear conclusion can be made due to insufficient evidence, but there was a possibility that topic identifiers had an undesirable influence on the accuracy of term rankings.

Table 7.3. The performance via $\mathrm{PMI}_{\mathrm{topic}}$ over the 'topic-replaced n-gram corpus'

	Geo. mean	Success@1	Success@10	MRR@150
Last experiment (non-redu.)	26.78	7.41%	31.48%	0.1459
Last experiment (positive)	44.70	1.85%	16.67%	0.0634
Excluded replaced n-grams	30.18	6.02%	26.39%	0.1249
Excluded repl. (positive)	46.67	2.78%	14.81%	0.0668
+ All replaced n-grams	26.74	7.41%	28.24%	0.1495
+ All repl. (positive)	26.75	4.17%	27.31%	0.1213
+ Replaced unigrams	33.90	4.63%	24.07%	0.1127
+ Repl. unigrams (positive)	40.45	2.28%	19.91%	0.0779
+ Replaced bi- and trigrams	29.63	6.02%	26.85%	0.1325
+ Repl. 2,3-grams (positive)	31.71	4.63%	23.61%	0.1080

Table 7.4. The performance via $\mathrm{PMI}_{\mathrm{doc}}$ over the 'topic-replaced n-gram corpus'

	Geo. mean	Success@1	Success@10	MRR@150
Last experiment (non-redu.)	22.83	14.35%	33.80%	0.2135
Last experiment (positive)	38.18	2.31%	18.98%	0.0801
Excluded replaced n-grams	22.37	14.35%	33.33%	0.2169
Excluded repl. (positive)	37.31	2.78%	19.44%	0.0847
+ All replaced n-grams	20.06	12.50%	36.57%	0.2094
+ All repl. (positive)	35.59	3.70%	20.83%	0.0927
+ Replaced unigrams	21.81	11.11%	33.80%	0.1952
+ Repl. unigrams (positive)	40.08	2.31%	17.59%	0.0743
+ Replaced bi- and trigrams	25.66	8.33%	31.48%	0.1684
+ Repl. 2,3-grams (positive)	38.44	2.78%	18.52%	0.0805

7.4.2 Document Correlation Scoring

Incorporating document correlation with topic replacement could improve the accuracy under certain conditions. The results are given in Table 7.3 and Table 7.4; and Figure 7.1, Figure 7.2 and Figure 7.3, where *Last experiment* means the results over the 'non-redundant n-gram corpus' shown in Chapter 6. In particular, when using topic-based PMI with an intercept, the performance was dramatically improved. For example, the approach with the topic-replaced corpus recorded a MMR@150 of 0.2511 under $T_{\text{intercept}} = 6.0$ in Figure 7.3 whereas with the 'non-redundant n-gram corpus', evaluated in the last experiment, under $T_{\text{intercept}} = 4.0$ this was only 0.2078 as in Figure 6.3. The origin of this improvement might be mainly due to the topic-replaced bigrams and trigrams. To clarify this reason, consider the four cases over the following test sets in Table 7.3 and Table 7.4: (1) topic-replaced n-grams were excluded, which was the same as the test set for *Last experiment*; (2) all topic-replaced terms were employed; (3) only topic-replaced unigrams were added; and (4) only topic-replaced bigrams and trigrams were added. Note that the training was run over the 'topic-replaced n-gram corpus', which includes all lengths of n-gram, regardless of what lengths of n-grams were employed when later evaluating. Then, the reason for the improvement was considered to be that in the case (4), the performance was similar to the case (2) rather than the case (3). Although unreplaced n-grams longer than a unigram (i.e. $n \geq 2$) led the system to poor answer re-ranking in the last experiment, the system trained with topic-replaced n-grams performed better. This was probably because such replaced n-grams were not sparse in comparison to unreplaced n-grams and positive correlation would become helpful for answer re-ranking as the sparsest corpus got the worst result in the last experiment with the document correlation scoring approach via topic-based PMI.

When taking into account only topic-replaced unigrams in the evaluation, the unigrams could not make a notable difference even though with an intercept there was a very slight improvement in performance. It seemed that topic-replaced n-grams offered an advantage by improving the performance with an intercept. In contrast, it was unique for the performance of the 'non-redundant n-gram corpus' to be improved by not employing an intercept. By adding topic-replaced unigrams, this tendency can be thought to have become weaker.

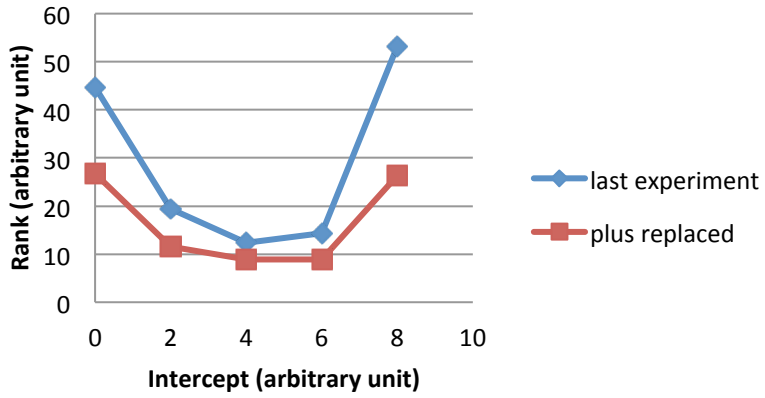

Figure 7.1. Geometric mean rank for various intercepts via topic-based PMI

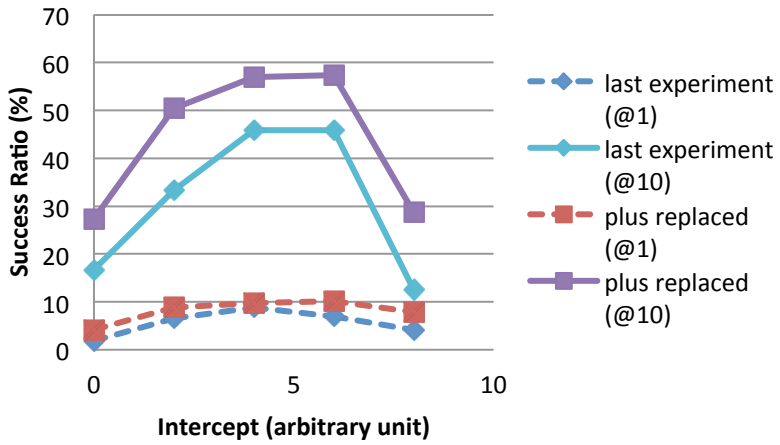

Figure 7.2. Success ratio for various intercepts via topic-based PMI

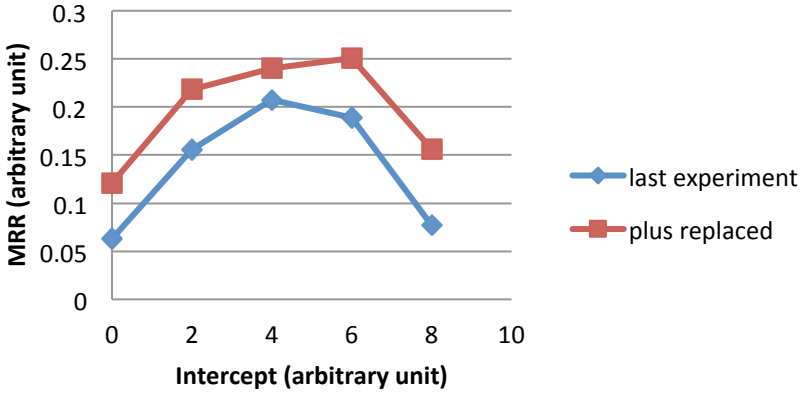

Figure 7.3. MRR@150 for various intercepts via topic-based PMI

Compared with the result for the 'non-redundant n-gram corpus' in Chapter 6, the results of an evaluation without topic-replaced n-grams showed worse performance even though the evaluation was run over the same set of vocabulary. This might be simply caused by the addition of topic-replaced n-grams into the corpus because the n-grams might behave as noise.

When it comes to document-based PMI, topic-replaced n-grams contributed to an improvement in the answer re-ranking system except the results for Success@1, and MRR@150, both without an intercept and using topic-replaced n-grams, as shown in Table 7.4. Interestingly, in the cases of both adding topic-replaced unigrams only and adding topic-replaced bigrams and trigrams, the performance was generally worse than adding no topic-replaced n-grams. As the 'non-redundant n-gram corpus' recorded the worst scores in Chapter 6, the subtraction of information, namely the exclusion of 2,3-grams and unigrams, respectively, might limit the accuracy of re-ranking.

An indistinct result was obtained in Table 7.5 when inferring over the 'topic-replaced unigram corpus'. Although there were no significant differences regardless of whether or not topic-replaced terms were employed in answer re-ranking, it can be said that topic-replaced unigrams slightly lowered the performance except in the case of using topic-based PMI only with positive PMI scores. This change was similar to the case of the performance for the 'topic-replaced n-gram corpus'.

Table 7.5. Performance when training over the 'topic-replaced unigram corpus'

	Geo. mean	Success@1	Success@10	MRR@150
Topic-based				
w/ replaced unigrams	54.28	1.39%	12.96%	0.0576
w/ repl. unigrams (positive)	34.34	1.85%	23.61%	0.0834
w/o replaced unigrams	52.16	2.78%	15.28%	0.0727
w/o repl. unigrams (pos)	45.29	1.39%	17.59%	0.0593
Document-based				
w/ replaced unigrams	9.53	14.35%	54.63%	0.2722
w/ repl. unigrams (positive)	23.81	4.63%	31.48%	0.1284
w/o replaced unigrams	8.32	23.15%	58.80%	0.3509
w/o repl. unigrams (pos)	21.99	7.41%	31.48%	0.1522

7.4.3 Query Expansion

Topic replacement could not improve the answer re-ranking system via query expansion for both topic-based PMI in Table 7.6 and document-based PMI in Table 7.7. These results should be compared to those of the no expansion model in Table 7.8. The fact that the performance of query expansion systems was strongly related to that of the no expansion system indicated that the deterioration was triggered by adding topic-replaced terms into the evaluation corpus regardless of the approach of query expansion. In other words, Okapi/BM25 did not suit evaluating topic-replaced terms directly. A possible reason was that topic-replaced term would attract irrelevant answer candidate documents since topic identifiers would be too general to detect the correct answer document when compared to original keywords, which tended to be specific.

The exaggerated sparseness of topic identifiers due to the non-redundant constraint might have influenced the scoring of answer documents via Okapi/BM25. For instance, the topic identifier of topic #1 occurred only 637 times in the 'topic-replaced n-gram corpus' whereas that occurred 1987 times if redundant shorter n-grams were not eliminated. The sparseness might have let topic-replaced terms be regarded as unreasonably important under Okapi/BM25 due to the IDF modifier. This would have been problematic because each question normally has a number of them. On the other hand, this sparseness seems not to be influential because the evaluation over the system whose training corpus consisted of redundant topic-replaced n-grams as well as unreplaced n-grams indicated the system performed similarly to the system based only on the 'topic-replaced n-gram corpus', where redundant shorter n-grams had been eliminated (although these figures are not shown in the thesis).

In terms of the extent of the improvement due to query expansion, topic-replaced terms could make a large difference for both topic-based PMI and document-based PMI, especially for the former, although worse performance tends to be easy to be improved. The 'topic-replaced n-gram corpus' showed a 27% better geometric mean rank than the no expansion model, utilising all topic-replaced n-grams, whereas it had only a 8.6% improvement utilising non-redundant topic-replaced n-grams.

Table 7.6. The performance for topic-based PMI (T=7.0, l=1000, c=0.2, h=3)

	Geo. mean	Success@1	Success@10	MRR@150
Last experiment (non-redu.)	4.375	33.80%	75.00%	0.4801
Excluded replaced n-grams	4.493	31.48%	75.46%	0.4597
+ All replaced n-grams	5.830	23.15%	67.59%	0.3733
+ Replaced unigrams	7.790	17.13%	61.11%	0.2991
+ Replaced bi- and trigrams	4.513	32.41%	75.00%	0.4573

Table 7.7. The performance for document-based PMI (T=7.0, l=1000, c=0.3, h=3)

	Geo. mean	Success@1	Success@10	MRR@150
Last experiment (non-redu.)	4.686	34.26%	73.15%	0.4732
Excluded replaced n-grams	4.723	33.80%	73.15%	0.4655
+ All replaced n-grams	7.620	15.28%	60.65%	0.3090
+ Replaced unigrams	9.444	11.11%	52.78%	0.2555
+ Replaced bi- and trigrams	5.023	30.09%	70.37%	0.4367

Table 7.8. The performance of no expansion model

	Geo. mean	Success@1	Success@10	MRR@150
Last experiment (non-redu.)	4.861	31.48%	73.15%	0.4547
Excluded replaced n-grams	4.915	31.02%	73.15%	0.4475
+ All replaced n-grams	8.036	12.50%	59.26%	0.2897
+ Replaced unigrams	9.986	9.72%	52.31%	0.2403
+ Replaced bi- and trigrams	5.257	27.31%	69.44%	0.4173

7.5 Further Discussion

This experiment revealed that topic replacement can bring a notable improvement for the document correlation scoring approach via topic-based PMI. Although the employment of an IDF modifier also can be listed for testing in future experiments, it is unlikely the modifier will work particularly well after reviewing the result of the query expansion model, where topic-replaced terms prevented high accuracy of the system.

When it comes to the query expansion approach, in order to use this with the topic replacement approach, consideration needs to be given to the infrequency of topic-replaced unigrams in the training corpus. The possible reason why topic-replaced terms degraded

the accuracy in the evaluation was that topic-replaced terms were too general to recognise which document was related. One of the possible measures is to lower the importance of topic-replaced terms when scoring the terms. However, a more sophisticated approach may be required to improve the system. For example, incorporating a pattern matching approach [12, 34 and 35] to form topic-replaced terms may have a positive influence if the approaches currently employed to form the terms were too simple.

7.6 Conclusion

Topic replacement has appealing qualities for the topic-based document correlation scoring approach. Employing intercepts and topic-based PMI could improve this approach from 0.2078 to 0.2511 in terms of MRR@150 although the improvement of the system via document-based PMI was not significant. On the other hand, the experiment using a query expansion model showed inferior accuracy to that for no expansion model. This result was not attributed to query expansion itself but it was caused by the mismatch between Okapi/BM25 and the topic replacement approach. Specifically, topic identifiers may be too general for such a system to select the correct answer documents.

8 Conclusion

This thesis explored a question answering (QA) system for answer re-ranking which incorporated a topic model, a bilingual Dirichlet allocation (Bi-LDA) model [1], and knowledge extracted from a social question answering (SQA) forum [2, 3, 4, 5, 6, 7, 8, 9 and 10]. Query expansion [83] was found to be effective when applied by the system.

The query expansion approach brought a 16% improvement in the top-150 mean reciprocal rank (MRR@150), compared to the QA system based on Okapi/BM25. When n-grams were taken into account, we demonstrated that they enhanced the accuracy of the answer re-ranking system. Also, in order to address the data sparseness problem, we examined the system with a topic-replaced corpus, wherein the topical terms were added after, or replaced with, the topic identifiers. Topic-replacement did not perform well with query expansion, although there were some benefits as well as the drawbacks.

The main finding of this research was that the Bi-LDA model could improve the answer re-ranking system by employing a query expansion approach, as shown in Chapter 5. Since an existing model proposed by Park et al. [83] did not work well with the Bi-LDA model in this research, we proposed alternative approaches based on pointwise mutual information (PMI) and succeeded in obtaining a better result. It can be considered that the improvement was due to the logarithmic scale of PMI, which treats each relationship more equally than a linear scale. In addition, the selection of expanded terms lead to a further improvement: specifically, we discarded the expanded terms which have relatively low scores under the assumption that they would not be important. Furthermore, since topic models are expected to perform better for topical terms, the division of the problem into topical and non-topical terms improved the model accuracy due to the complementary disadvantages of each category.

We further showed in Chapter 6 that n-grams could enhance the accuracy of the answer re-ranking system based on query expansion. However, it would be problematic for all the n-grams to be added to a corpus regardless of their length because the corpus then guarantees the co-occurrence of an n-gram and all shorter n-grams, which are part of the original n-gram. This would mean the original n-gram and the shorter ones were assigned to the same topic, contrary to the intention of employing n-grams. We demonstrated that by employing

only the original n-grams – namely by excluding the shorter n-grams – the system with the query expansion approach achieved more accurate performance. In addition, the weighting function defined in Chapter 6 contributed to the improvement in this experiment; this function tended to treat infrequent terms as important. By combining this tendency with the empirical knowledge that essential keywords are typically infrequent, the weighting function allowed n-grams to work well.

On the other hand, the topic-replacement approach could not improve the system with query expansion (as shown in Chapter 7). This is probably because topic-replaced terms are too general to successfully recognise the correct answer documents out of the candidates. However, since it is essential for QA systems to solve the data sparseness problem, the topic-replacement approach should be developed further and made more sophisticated. When using PMI scores to measure similarity between a question and the answer candidates for answer re-ranking, the topic-replacement approach performed better than without topic-replacement. This might be a consequence of the higher frequency of the topic-replaced terms than the original terms, namely the alleviation of the data sparseness problem.

Many further improvements are required to develop a practical QA system. However, this research importantly revealed that a Bi-LDA model could improve an answer re-ranking system based on query expansion with a large corpus extracted from a SQA forum.

References

1. I. Vulić, W. D. Smet and M.-F. Moens, “Cross-language information retrieval models based on latent topic models trained with document-aligned comparable corpora”, *Information Retrieval*, Vol. 16 (3), June 2013, pp. 331-368.
2. J. Bian, Y. Liu, E. Agichtein and H. Zha, “Finding the right facts in the crowd: factoid question answering over social media”, in *WWW '08 Proceedings of the 17th international conference on World Wide Web*, April 2008, pp. 467-476.
3. Y. Liu, S. Li, Y. Cao, C.-Y. Lin, D. Han and Y. Yu, “Understanding and summarizing answers in community-based question answering services”, in *COLING '08 Proceedings of the 22nd International Conference on Computational Linguistics*, Vol. 1, August 2008, pp. 497-504.
4. Y. Liu, J. Bian and E. Agichtein, “Predicting information seeker satisfaction in community question answering”, in *SIGIR '08 Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, July 2008, pp. 483-490.
5. X. Xue, J. Jeon and W. B. Croft, “Retrieval models for question and answer archives”, in *SIGIR '08 Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, July 2008, pp. 475-482.
6. S. Li and S. Manandhar, “Improving question recommendation by exploiting information need”, *HLT '11 Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, June 2011, pp. 1425-1434.
7. T. Mori, T. Okubo and M. Ishioroshi, “A QA system that can answer any class of Japanese non-factoid questions and its application to CCLQA EN-JA task”, in *Proceedings of NTCIR-7 Workshop Meeting*, December 2008, pp. 41-48.
8. Y. Wu and H. Kawai, “Exploiting Social Q&A Collection in Answering Complex Questions”, in *Proceedings of the Joint Conference on Chinese Language Processing (CLP2010)*, August 2010.
9. M. Surdeanu, M. Ciaramita and H. Zaragoza, “Learning to rank answers to non-factoid questions from web collections”, *Computational Linguistics*, Vol. 37 (2), June 2011, pp. 351-383.

10. K. Wang, Z. Ming and T.-S. Chua, “A syntactic tree matching approach to finding similar questions in community-based qa services”, in *SIGIR '09 Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, July 2009, pp. 187-194.
11. H. Menzel, “Information Needs and Uses in Science and Technology”, *Annual Review of Information Science and Technology*, Vol. 1, Interscience Publishers, 1966, pp. 41-69.
12. D. Ravichandran and E. Hovy, “Learning Surface Text Patterns for a Question Answering System”, in *ACL '02 Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, July 2002, pp. 41-47.
13. D. M. Blei, A. Y. Ng and M. I. Jordan, “Latent Dirichlet allocation”, *The Journal of Machine Learning Research*, Vol. 3, March 2003, pp. 993-1022.
14. T. L. Griffiths and M. Steyvers, "Finding scientific topics", in Proceedings of the National Academy of Sciences, Vol. 101 (1), April 2004, pp. 5228-5235.
15. Y. W. Teh, D. Newman and M. Welling, "A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation", *Advances in Neural Information Processing Systems (NIPS)*, Vol. 19, 2007.
16. A. Asuncion, M. Welling, P. Smyth and Y. W. Teh, "On smoothing and inference for topic models", in *UAI '09 Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, June 2009, pp. 27-34.
17. B. F. Green Jr., A. K. Wolf, C. Chomsky, and K. Laughery, “BASEBALL: an Automatic Question Answerer”, in *IRE-AIEE-ACM '61 (Western) Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, May 1961, pp. 219-224.
18. T. Winograd, “Understanding Natural Language”, *Cognitive Psychology*, Vol. 3 (1), January 1972, pp. 1–191.
19. Wikipedia, “SHRDLU – Wikipedia, the free encyclopedia” [Online]. Available: http://en.wikipedia.org/wiki/SHRDLU [Last Modified: 14 March 2013, 18:14].
20. H. Isozaki, R. Higashinaka, M. Nagata and T. Kato, supervised by M. Okumura, *Question Answering System*, Tokyo: Corona Publishing Co., LTD., 2009 (in Japanese).

21. J. Suzuki, Y. Sakaki and E. Maeda, “SVM Answer Selection for Open-domain Question Answering”, in *COLING '02 Proceedings of the 19th international conference on Computational linguistics*, September 2002, Vol. 1, pp. 1-7.
22. D. E. Appelt, J. R. Hobbs, J. Bear, D. J. Israel and M. Tyson, FASTUS: A Finite-state Processor for Information Extraction from Real-world Text, in *IJCAI 1993*, August 1993, pp. 1172-1178.
23. J. Suzuki, Y. Sasaki and E. Maeda, “Question Type Classification Using Word Attribute AT-gram and Statistical Machine Learning (Natural Language Processing)”, *IPSJ Journal*, Vol. 44 (11), November 2003, pp. 2839-2853 (in Japanese).
24. Wikipedia, “tf-idf – Wikipedia” [Online]. Available: http://ja.wikipedia.org/wiki /Tf-idf [Last Modified: 20 March 2013, 4:15] (in Japanese).
25. A. Aizawa, "An information-theoretic perspective of tf–idf measures", Information Processing and Management, Vol. 39, 2003, pp. 45–65.
26. K. S. Jones, S. Walker and S. E. Robertson, “A probabilistic model of information retrieval: development and comparative experiments”, *Information Processing and Management*, Vol. 36 (6), November 2000, pp. 779-808.
27. H. Isozaki, “An Analysis of a High Performance Japanese Question Answering System”, *Journal ACM Transactions on Asian Language Information Processing*, Vol. 4 (3), September 2005, pp. 263-279.
28. A. Berger, R. Caruana, D. Cohn, D. Freitag and V. Mittal, “Bridging the Lexical Chasm: Statistical Approaches to Answer-Finding”, in *SIGIR '00 Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, July 2000, pp. 192-199.
29. J. Prager, E. Brown, A. Coden and D. Radev, “Question-answering by predictive annotation”, SIGIR, ACM, NY, USA in *SIGIR '00 Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, July 2000, pp. 184-191.
30. R. Grishman and B. Sundheim, “Message Understanding Conference-6: a brief history”, in *COLING '96 Proceedings of the 16th conference on Computational linguistics*, Vol. 1, August 1996, pp. 466-471.
31. S. Sekine and H. Isahara, “IREX: IR and IE evaluation project in Japanese”, in *LREC 2000*, No. 27, May 2000.

32. J. Andrew and A. J. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm”, in *IEEE Transactions on Information Theory*, Vol. 13 (2), April 1967, pp. 260–269.
33. J. D. Lafferty, A. McCallum and F. C. N. Pereira, “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data”, in *ICML '01 Proceedings of the Eighteenth International Conference on Machine Learning*, June 2001, pp. 282-289.
34. P. Pantel and D. Ravichandran, “Automatically Labeling Semantic Classes”, in *HLT/NAACL 2004*, May 2004, pp. 321-328.
35. H. Cui, K. Li, R. Sun, T.-S. Chua and M.-Y. Kan, “National University of Singapore at the TREC 13 Question Answering Main Task”, in *Proceedings of the 13th TREC*, November 2005.
36. M. Collins and N. Duffy, “Convolution Kernels for Natural Language”, in *Proceedings of the 14th Conference on Neural Information Processing Systems*, 2001.
37. K. Toutanova and C.D. Manning, “Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger”, in *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 2000, pp. 63-70.
38. R. Socher, J. Bauer, C. D. Manning and A. Y. Ng, “Parsing With Compositional Vector Grammars”, in *Proceedings of ACL 2013*, 2013.
39. T. Hirano, Y. Matsuo and G. Kikui, “Detecting semantic relations between named entities in text using contextual features”, in *ACL '07 Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, June 2007, pp. 157-160.
40. C. L. A. Clarke and E. L. Terra, “Passage Retrieval vs. Document Retrieval for Factoid Question Answering”, in *SIGIR '03 Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, July 2003, pp. 427-428.
41. C. L. A. Clarke, G. V. Cormack and T. R. Lynam, “Exploiting Redundancy in Question Answering”, in *SIGIR '01 Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, September 2001, pp. 358-365.

42. E. Brill, J. Lin, M. Banko, S. Dumais and A. Ng, “Data-Intensive Question Answering”, in *Proceedings of the Tenth Text REtrieval Conference*, 2001, pp. 393-400.
43. G. Adomavicius and A. Tuzhilin, “Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions”, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17 (6), June 2005, pp. 734-749.
44. D. I. Moldovan, C. Clark, S. Harabagiua and S. Maiorano, “COGEX: A Logic Prover for Question Answering”, in *NAACL '03 Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Vol. 1, May 2003, pp. 87-93.
45. E. M. Voorhees, “Overview of the TREC 2002 Question Answering Track”, in *Proceedings of the Eleventh Text REtrieval Conference*, 2002, pp. 115-123.
46. D. I. Moldovan, C. Clark and M. Bowden, “Lymba’s PowerAnswer 4 in TREC 2007” in *Proceedings of the Sixteenth Text REtrieval Conference*, 2007.
47. H. T. Dang, D. Kelly and J. Lin, “Overview of the TREC 2007 Question Answering Track”, in *Proceedings of the Sixteenth Text REtrieval Conference*, 2007.
48. M. Tatu and D. I. Moldovan, “COGEX at RTE 3”, in *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, June 2007, pp. 22-27.
49. C. C. Green, “Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing”, *Machine Intelligence*, Vol. 4, edited by B. Meltzer and D. Michie, 1969, pp. 183-205.
50. W3C. “RDF – Semantic Web Standards” [Online]. Available: http://www.w3.org /RDF/, [Accessed at: 28 March 2013, 19:31].
51. C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*, Cambridge: MIT Press, 1999.
52. S. Coates-Stephens, “Automatic Acquisition of Proper Noun Meanings”, in *Proceedings of the 6th International Symposium on Methodologies for Intelligent Systems*, October 1991, pp. 306-315.
53. M. A. Hearst, “Automatic acquisition of hyponyms from large text corpora”, in *COLING '92 Proceedings of the 14th conference on Computational linguistics*, Vol.2, August 1992, pp. 539-545.

54. H. Joho and M. Sanderson, “Retrieving descriptive phrases from large amounts of free text”, *Information Processing and Management*, Vol. 36, 2000, pp. 779-840.
55. B. Katz, J. J. Lin, D. Loreto, W. Hildebrandt, M. W. Bilotti, S. Felshin, A. Fernandes, G. Marton and F. Mora, “Integrating Web-based and corpus-based techniques for question answering”, in *Proceedings of the Twelfth Text REtrieval Conference*, 2003, pp. 426-435.
56. E. M. Voorhees, “Overview of the TREC 2003 Question Answering Track”, in *Proceedings of the Twelfth Text REtrieval Conference*, 2003, pp. 54-68.
57. D. Lin and D. Demner-Fushman, “Will Pyramids Built of Nuggets Topple Over?”, in *HLT-NAACL '06 Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, June 2006, pp. 383-390.
58. D. Lin and D. Demner-Fushman, “Automatically Evaluating Answers to Definition Questions”, in *HLT '05 Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, October 2005, pp. 931-938.
59. J. Xu, A. Licuanan and R. M. Weishedel, “TREC 2003 QA at BBN: Answering Definitional Questions”, in *Proceedings of the Twelfth Text REtrieval Conference*, 2003, pp. 98-106.
60. H. Yang, H. Cui, M. Maslennikov, L. Qiu, M.-Y. Kan and T.-S. Chua, “QUALIFIER In TREC-12 QA Main Task”, in *Proceedings of the Twelfth Text REtrieval Conference*, 2003, pp. 480-488.
61. Y. Zhou, X. Yuan, J. Cao, X. Huang and L. Wu, “FDUQA on TREC2006 QA Track”, in *Proceedings of the Fifteenth Text REtrieval Conference*, 2006, pp. 480-488.
62. J. Prager, J. Chu-Carroll, K. Czuba, C. Welty, A. Ittycheriah and R. Mahindru, “IBM’s PIQUANT in TREC2003”, in *Proceedings of the Twelfth Text REtrieval Conference*, 2003, pp. 283-292.
63. U. Shibusawa, T. Hayashi and R. Onai, ‘Development and Evaluation of a System for Extracting Answers of a "Why" Type Question from the WEB’, *IPSJ Journal*, Vol. 48 (3), March 2007, pp. 1512-1523 (in Japanese).
64. M. Selfridge, J. Daniell and D. Simmons, “Learning Causal Models by Understanding Real-World Natural Language Explanations”, *in CAIA 1985*

Proceeding of the Second Conference, Artificial Intelligence Applications, December 1985, pp. 378-383.

65. R. Girju, "Automatic detection of causal relations for Question Answering", in *MultiSumQA '03 Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering*, Vol. 12, 2003, pp. 76-83.
66. K. Morooka and J. Fukumoto, Answer Extraction Method for Why-type Question Answering System", *IEICE technical report. Natural language understanding and models of communication*, Vol. 105 (594), January 2006, pp. 7-12 (in Japanese).
67. R. Higashinaka and H. Isozaki, "Corpus-based Question Answering for why-Questions", in *Proceedings of IJCNLP*, January 2008, pp. 418-182.
68. H. Isozaki and R. Higashinaka, "Web NAZEQA, a Web-based Why-Question Answering System", *IPSJ Journal*, 2008, pp. 143-146 (in Japanese).
69. S. Verberne, L. Boves, N. Oostdijk and P.-A. Coppen, "What Is Not in the Bag of Words for *Why*-QA?", in *Computational Linguistics*, Vol. 36 (2), June 2010, pp. 229-245.
70. Y. Tamura, J. Murakami, M Tokuhisa and S. Ikehara, "A Study for the Why-type Question-Answering System using Web Search Engine", *Study Report of IPSJ 2008*, Vol. 4, January 2008, pp. 15-21 (in Japanese).
71. J.-H. Oh, K. Torisawa, C. Hashimoto, T. Kawada, S. D. Saeger, J. Kazama and Y. Wang, "Why Question Answering using Sentiment Analysis and Word Classes", in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, July 2012, pp. 368-378.
72. N. Asanoma, O. Furuse and R. Kataoka, "Feature Analysis of Explanatory Documents for How-to Type Question Answering", *Study Report of IPSJ 2005*, Vol. 73, July 2005, pp. 55-60 (in Japanese).
73. H. T. Dang, "Overview of the TAC 2008, Opinion Question Answering and Summarization Tasks" [Online], Available: http://www.nist.gov/tac/publications
    a. /2008/presentations/TAC2008_Opinion_overview.pdf, [Accessed: 31 March 2013, 12:36].
74. F. Li, Z. Zheng, Y. Tang, F. Bu, R. Ge, X. Zhu and X. Zhang, "THU QUANTA at TAC 2008 QA and RTE track", in TAC 2008, November 2008.
75. F. Li, Y. Tang, M. Huang and X Zhu, "Answering Opinion Questions with Random Walks on Graphs", *ACL '09 Proceedings of the Joint Conference of the 47th*

Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Vol. 2, August 2009, pp. 737-745.

76. L. Page, S. Brin, R. Motwani and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web", *Technical report, Stanford University*, 1999.
77. I. Dagan, B. Dolan, B. Magnini and D. Roth, "Recognizing textual entailment: Rational, evaluation and approaches", *Natural Language Engineering*, Vol. 15 (4), October 2009, pp. i-xvii.
78. A. Peñas, E. H. Hovy, P. Forner, Á. Rodrigo, R. F. E. Sutcliffe, C. Forascu and C. Sporleder, "Overview of QA4MRE at CLEF 2011: Question Answering for Machine Reading Evaluation", in *Proceeding of CLEF 2011 Labs and Workshop, Notebook Papers*, September 2011, pp. 19-22.
79. F. M. Zanzotto, M. Pennacchiotti and A. Moschitti, "A machine learning approach to textual entailment recognition", *Natural Language Engineering*, Vol. 15 (4), October 2009, pp. 551-582.
80. J. Bos and K. Markert, "Recognising textual entailment with logical inference", in *HLT '05 Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, April 2005, pp. 628-635.
81. A. Celikyilmaz, D. Hakkani-Tur and G. Tur "LDA Based Similarity Modeling for Question Answering", in *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search*, Jun 2010, pp. 1-9.
82. M. Liu, Y. Liu and Q. Yang, "Predicting Best Answerers for New Questions in Community Question Answering", *Web-Age Information Management, Lecture Notes in Computer Science*, Vol. 6184, 2010, pp 127-138
83. L. A. F. Park and K. Ramanohanarao, "Efficient storage and retrieval of probabilistic latent semantic information for information retrieval", *The VLDB Journal*, Vol. 18, 2009, pp. 141-155.
84. T. Hofmann, "Probabilistic latent semantic indexing", in *SIGIR '99 Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, August 1999, pp 50-57.
85. S. T. Dumais, "Latent semantic analysis", *Annual Review of Information Science and Technology*, Vol. 38 (1), 2004, pp. 188-230.
86. D. Andrzejewski, "Expectation Maximization", available online: http://pages.cs.wisc.edu/~andrzeje/research/em.pdf, Feb 2010.

87. M. Abramowitz and I. Stegun, editors, *Handbook of Mathematical Functions*, Dover, New York, 1970.
88. R. A. Beaza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley Longman Publishing Co., Inc., Boston MA, USA, 1999.
89. S. Verberne, L. Boves, N. Oostdijk, P.-A. Coppen, "Evaluating discourse-based answer extraction for why-question answering", in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, July 2007, pp. 735-736.