Using Linguistically Motivated Features in Document Retrieval for Question Answering

Luiz Augusto Sangoi Pizzato
M.Sc., B.CompSc.

This thesis is presented for the degree of
Doctor of Philosophy

Department of Computing
Faculty of Science
Macquarie University

July 2009

Contents

Abstract

This thesis investigates the impact of using linguistic features in the Information Retrieval (IR) stage of Question Answering (QA) systems. We hypothesise that techniques that are commonly used in the final answer extraction stage can improve the overall results of a QA system when adopted in the earlier IR stage. In particular, we study the use of the following information: i) named entities in a pseudo-relevance feedback process; and ii) semantic relations between words of questions and text sentences.

The study of the use of named entities is inspired by the common practice of filtering out sentences that do not contain the expected answer type. We consequently introduce a pseudo-relevance feedback that inserts entities of the correct answer type in the original query. Our experiments show that this technique leads to a query drift and the final results do not improve with respect to a query without additional feedback.

To study the use of relational information, we design an IR framework that is more efficient (in both speed and memory consumption) than standard approaches based on relational databases and on the concatenation of word pairs at the indexing stage. The resulting framework allows a multilayer index that uses an extension to the standard vector space model as a ranking strategy. The resulting ranking strategy improves precision, without compromising the overall recall, by including linguistic word relations.

We present the QPLM, a model of relational information that borrows concepts from Semantic Role Labelling (SRL) but is designed for the fast generation of annotation and its use for indexing and retrieval. The results are of quality comparable to SRL and indicate that linguistic information encoded in the form of semantic relations does enhance the retrieval quality of text and the final accuracy of QA systems.

Statement of Candidate

Statement of Candidate

I certify that the work in this thesis entitled "Using Linguistically Motivated Features in Document Retrieval for Question Answering" has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree to any other university or institution other than Macquarie University.

I also certify that the thesis is an original piece of research and it has been written by me. Any help and assistance that I have received in my research work and the preparation of the thesis itself have been appropriately acknowledged.

In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature:

Luiz Augusto Sangoi Pizzato - 40452239

Sydney, 20 July, 2009

Acknowledgements

I would like to acknowledge that this thesis would not be otherwise possible without the financial assistance of the iMURS scholarship from Macquarie University and CSIRO, and without the scholarship from the ICS department in this last year.

I would like to express my gratitude to my supervisor Dr. Diego Mollá for the guidance throughout all these years and for giving me the motivation needed to complete the hard work involved in a PhD thesis. I am highly indebted to him for his support and belief in me.

I would like to thank my co-supervisors Dr. Cécile Paris, who with her expertise provided me with invaluable feedback that helped to shape this research, and Dr. Rolf Schwitter, who supervised me during Diego's sabbatical and gave me important suggestions for this final document.

I would like to say *muito obrigado* to my family, who even though were geographically distant for the whole duration of my studies, were never far away from my heart.

Finally, I would like to express my greatest thanks to my wife Joanne. She is the most important person in my life, whose love and support gave me the emotional balance so much needed during all the ups and downs of the PhD. Joanne is a superwoman, who not only worked full-time to support the two of us during this final year, but also made time to read and re-read all the thesis drafts for me.

Joanne, if I could I would give you anything in the universe, but the only thing that I have is my love for you and this thesis, which I dedicate to you.

Publications Arising from this Work

The research described in this thesis produced a number of peer-reviewed publications. The following publications detail the trie-based classification method described in Chapter 3.

(i) Luiz Augusto Sangoi Pizzato. Using a trie-based structure for question analysis. In *Proceedings of the Australasian Language Technology Workshop 2004*, pages 25–31, Macquarie University, Sydney, Australia, 2004. ASSTA.

(ii) Menno van Zaanen, Luiz Augusto Pizzato, and Diego Mollá. Question classification by structure induction. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, Edinburgh, UK, pages 1638–1639. 2005.

(iii) Menno van Zaanen, Luiz Pizzato, and Diego Mollá. Classifying sentences using induced structure. In *Proceedings of the Twelfth Edition of the Symposium on String Processing and Information Retrieval (SPIRE2005)*, Buenos Aires, Argentina, pages 139–150. 2005.

The following publication describes the QA system named MetaQA, which is used for the evaluation of the retrieved set of documents in Chapter 6.

(iv) Luiz Augusto Pizzato and Diego Mollá. Extracting exact answers using a meta question answering system. In *Proceedings of the Australasian Language Technology Workshop 2005*, pages 105–112, Sydney, Australia, 2005.

The following publication describes the pseudo-relevance feedback technique detailed in Chapter 3.

(v) Luiz Pizzato, Diego Mollá, and Cécile Paris. Pseudo-relevance feedback using named entities for question answering. In *Proceedings of the Australasian Language Technology Workshop*, volume 4, pages 83–90, Sydney, Australia, 2006.

The following publications describe the work related to QPLM (Chapter 5), the IR framework (Chapter 4) and its evaluation (Chapter 6).

(vi) Luiz Augusto Pizzato and Diego Mollá. Question prediction language model. In *Proceedings of the Australasian Language Technology Workshop 2007*, pages 92–99, Melbourne, Australia, 2007.

(vii) Luiz Augusto Pizzato and Diego Mollá. Indexing on semantic roles for question answering. In *Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question Answering*, pages 74–81, Manchester, UK, 2008.

During the PhD studies I was part of the AnswerFinder (AF) research group that participated at the QA tracks at the Text REtrieval Conference (TREC). The following publication originated from our participation in 2006.

(viii) Menno van Zaanen, Diego Mollá, and Luiz Pizzato. AnswerFinder at TREC 2006. In *The Fifteenth Text REtrieval Conference (TREC 2006)*, Gaithersburg, USA, 2006.

List of Acronyms

IR	Information Retrieval
QA	Question Answering
WWW	World Wide Web
NLP	Natural Language Processing
TREC	Text REtrieval Conference
MAP	Mean Average Precision
MRR	Mean Reciprocal Rank
EAT	Expected Answer Type
AF	AnswerFinder
SRL	Semantic Role Labelling
POS	Part-of-Speech
BoW	Bag-of-Words
TF	Term Frequency
IDF	Inverted Document Frequency
QPLM	Question Prediction Language Model
NYT	The New York Times News Service
APW	The Associated Press Worldstream News Service
XIE	The Xinhua News Service

1 Introduction

The amount of digital information available today is extremely large, and with the advent of computers and fast distributed networks, we have the possibility to access this information at any given moment. However, large repositories of data such as the World Wide Web (WWW) are only truly usable if there is a means to filter the information to only what is needed by the user. The process of filtering information according to the user's need is called Information Retrieval (IR). Different terms are applied to describe different types of IR processes: Document Retrieval is used when users are looking for documents, Image Retrieval when users are looking for images, and Question Answering (QA) when users formulate questions and expect answers. In this thesis, we focus on the search for information within the context of QA.

This introductory chapter presents some basic concepts regarding IR (Section 1.1) and QA (Section 1.2). A basic understanding of these disciplines is an important prerequisite for Section 1.3, where we introduce the research questions on which this thesis is based. The last part of this chapter presents the structure for this thesis.

1.1 Information Retrieval (IR)

IR is a discipline that is highly visible today due to the popularity of cur-

rent Internet search engines such as Yahoo! Search[1], Microsoft Live[2], and Google[3]. IR is also an essential task to help users to deal with information overload, caused by large amounts of data that is created and stored on a regular basis. Manning et al. [86, Chapter 1] define IR in the following way:

> "*Information Retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).*"

IR deals with the problem of finding information that is relevant to a user's need in a large repository of information. The information can consist of any type of media such as text, audio, image, and video. In the scope of this thesis, we restrict the definition of IR to the *processing of text documents so that an information need, expressed by a set of words (the query), is satisfied by a set of text documents or by parts of these documents.*

The success of the IR process is measured by the relevance of the retrieved text according to the presented query. The query, which is often a set of disconnected words, represents a rather vague statement of the information need. For instance, the query "*Prime Minister of Australia*" is clear on what needs to be searched, but does not specify what is the actual information need of the user, which makes it hard to judge documents for their relevance. A better way to express the information need is by the use of natural language questions. For instance, the question "*Who is the Prime Minister of Australia?*" represents a clear need of information that can be easily used to judge the relevance of the retrieved documents. The task of finding answers to natural language questions is called Question Answering.

1.2 Question Answering (QA)

QA is able to better handle some of the problems of keyword-based search,

[1] `http://search.yahoo.com`
[2] `http://www.live.com`
[3] `http://www.google.com`

including: countering the low explicitness of the information need with the use of explicit sentences and questions in natural language; and facilitating the judgement of relevance by the user, who typically needs to read large portions of documents and sometimes perform in-document search to be able to determine whether the requested information is found. These two issues are even more salient when using devices with visual constraints such as portable communication devices or user interfaces for the visually impaired.

It has been long argued that natural language provides an easier way for users to query an IR system (e.g. [113]). However, we have shown in [109] that even when the search interface encourages the users to post queries as natural language questions, only a small percentage of entered queries are in that format. This is perhaps due to the high popularity of web search engines such as Google, which users have learned to use as a keyword-matching tool. With the sheer amount of WWW documents, search engines are able to successfully handle keyword-matching queries. This creates a great trust in these tools, despite the user having misconceptions on how IR tools work [99], and often requiring several queries to find the desired information [108]. As shown by Pan et al. [108], when more complicated information needs are required and the search engines fail to satisfy these needs, the users uphold the trust in these tools and shift the fault away from the search engine towards themselves for being unable to formulate the 'correct' query.

The research presented in this thesis brings the responsibility back to the IR task. We focus on the IR task and whether it can find documents that contain answers to natural language questions. By having an IR system that can identify documents that satisfy users' questions, it is likely to improve QA performance.

Most QA systems follow a standard three-stage architecture where questions are first analysed, then used in an IR system, and then finally matched with the retrieved documents in order to find answer candidates. Figure 1.1 illustrates the traditional QA pipeline architecture.

The question analysis stage deals with finding what type of information

Figure 1.1: Traditional QA pipeline architecture

is requested by the user and what is the topic of the question. The answer extraction is normally the last stage of a QA system, which focuses on finding the correct answer for a question in a selection of textual items. The question analysis and the answer extraction stages are where the most complex Natural Language Processing (NLP) techniques are applied, which include named-entity recognition [129], syntactic dependency parsing [28], Semantic Role Labelling [131], and automated theorem proving [93].

The IR stage is treated as a black box in the QA process and it is typically performed by off-the-shelf IR tools, which normally do not employ NLP techniques. Some QA systems focus on the input and output of IR systems; for example, Dumais et al. [35] perform a passive-to-active voice transformation of the question in an attempt to bring the IR query closer to the document that it is expected to retrieve. Some IR work focuses on improving QA by passage retrieval re-ranking using word overlap measures. For instance, Tellex et al. [144] compare a group of passage retrieval techniques and conclude that those that apply density-based metrics[4] are the most suitable to be used for QA.

Some work has been done on IR models that specifically aid the QA task. The work of Monz [98] defines a weighting scheme that takes into consideration the distance of the query terms. Murdock and Croft [100] propose a translation language model that defines the likelihood of the question being the translation of a specific document. Cui et al. [29] uses dependency relations over word density methods for passage retrieval. Tiedemann [147] uses a multi-layer index containing more linguistically-oriented information and a genetic learning algorithm to determine the best parameters for querying those indices when applied to the QA task. Tiedemann argues that since

[4]Ranking of passages based on the number of query words and the proximity between them.

QA is an all-natural language task, linguistically-oriented IR will help to find better documents for QA. However, the use of extra information may not necessarily improve QA alone; for instance, the high precision approach for IR of Bilotti and Nyberg [7] improves the IR results for QA, but fails to improve the final QA results.

Although IR can be helped by NLP techniques such as named-entity recognition, phrase extraction and syntax parsing [139], these techniques are not generally used due to their high complexity. Because IR has a history of high response times and fast speeds, any addition to the IR process, even if it improves the overall results, is not viewed upon favourably. On the other hand, QA is an IR dependant task that highly utilises NLP tools, which are normally an order of magnitude slower than IR. The addition of linguistic information in search indices, even though it can represent an added cost to the IR process, may actually reduce the overall cost of the QA process; for specific QA setups, the addition of such information may even speed up the QA experience as described by Bilotti et al. [10].

1.3 Research Question

IR should not solely aim to provide documents that match the exact keywords of a query, but should also perceive the user's information need and to find documents that can fulfil this need. QA is one attempt to make the information need more explicit by the use of natural language questions. In this thesis, we hypothesise that widely-used NLP information in QA, such as named entities, dependency relations and semantic role labels, can be used to assist the IR task in a QA environment. Specifically, we address the following research questions in this thesis:

1. *Can the presence of answer features such as named entities be used as an indication of relevance for documents in IR for QA?*

2. *Are syntactic dependencies and semantic role labels better indications of relevance of documents for QA than words alone? If so, can the*

same improvements be sustained with simpler semantic annotation?

This thesis addresses the issue of whether modifications in the IR structure, such as the addition of QA techniques including linguistic information, can help to fulfil the explicit information need of natural language questions and therefore be beneficial to QA. Our hypothesis is that QA can benefit from a specialised IR strategy that takes into account the natural language characteristics of the task. We hypothesise that if answer clues are included in the IR stage, better documents will be retrieved and therefore QA results will improve.

To test our hypotheses, we have experimented with different models of retrieval and information from NLP tools that are used in the QA task. We have experimented with IR using pseudo-relevance feedback, and with the inclusion of linguistic information in the search structure.

The addition of such information in the search structure is made possible by the implementation of an IR framework, which is another outcome of this thesis. The IR framework allows the inclusion of a vast range of relational information, and allows the construction of ranking mechanisms that are better targeted to the QA task. In a comparison with two off-the-shelf tools, the framework is more efficient in both speed and memory usage when using linguistic information.

1.4 Thesis Organisation

This thesis is organised into seven chapters, including this introduction. **Chapter 2** presents some background work in the field of IR and QA that are important for the understanding of this thesis. In addition, Chapter 2 presents some directly related work, which is discussed in the last section of the chapter. **Chapter 3** focuses on our first research question of whether named entities are a good indication of relevance for QA. In Chapter 3, we study a two-way IR architecture that considers the named-entity types of the expected answers in a pseudo-relevance feedback mechanism. Chapters 4 to 6

address our second research question of whether linguistic relations can provide a better indication of relevance for documents in IR than words alone. Because linguistic relations are poorly supported by IR tools, **Chapter 4** proposes an IR framework for the inclusion of linguistic information in the search structures and a novel ranking mechanism using this information. In **Chapter 5**, we define an annotation scheme that is based on the semantics of relationships between sentence words. This annotation scheme forms the basis to address our second research question, which is that when holding the assumption that linguistic information improves IR for QA, whether simpler semantic annotations are able to sustain these improvements. The proposed annotation scheme in Chapter 5 is faster to generate than current semantic annotation standards, which allows such techniques to be integrated into traditional speed critical phases such as IR. **Chapter 6** presents an evaluation of the usage of linguistic information, including our semantic annotation scheme, in the proposed IR framework. Finally, **Chapter 7** presents the concluding thesis remarks and future work.

2

Literature Review

The main focus of this thesis is to investigate techniques to improve the IR stages of a QA system. This chapter's objective is to introduce the main concepts that need to be understood for the full comprehension of the work described in the subsequent chapters. Section 2.1 and Section 2.2 focus on relevant background work in the IR and the QA areas. Section 2.3 further discusses some developments in IR that are specific to the QA task. Work that is more specifically related to the content of this thesis is presented in Section 2.4, which include the work of Tiedemann [146], and Bilotti and Nyberg [7] which use linguistic information to improve the IR for QA; Litkowski [77] which describes a simpler semantic relation model to extract answers to questions; and Fuhr and Gövert [43] which describes an efficient indexing method for structural retrieval.

2.1 Information Retrieval

According to Baeza-Yates and Ribeiro-Neto [4], IR deals with the representation, storage, organisation of, and access to information items. Information items can be seen as any type of information, such as text, image, audio, and video. However, because of the specific nature of this thesis, we will not deal with information other than textual documents.

When the number of documents is too great to be searched using full-

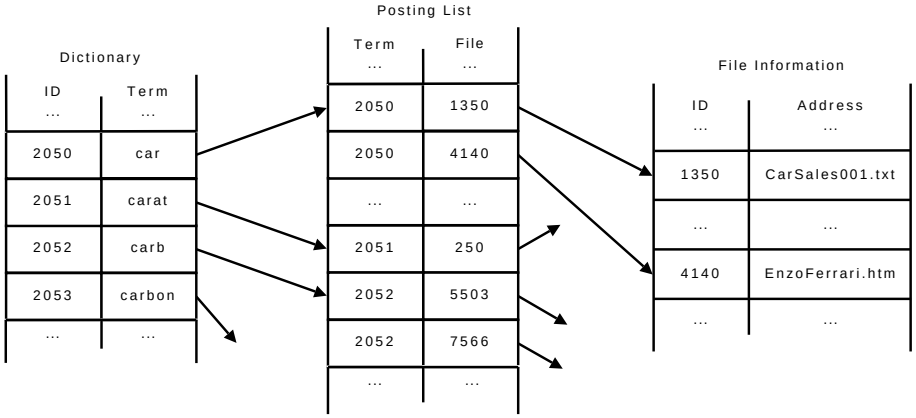


Figure 2.1: Example of the organisation of an inverted file

text scanning, the representation, storage and organisation of information become key aspects of the IR process. They allow the information to be readily accessible in case there is a need to retrieve it. Some representations of document collections allow quick access and retrieval of information. The most common are inverted files and signature files [37]. In our work, we will focus on inverted files, which according to Zobel [178] are a representation superior to signature files in terms of speed, space and functionality.

Inverted files are also known as inverted indices in the literature; in this thesis, we will use the term inverted files because this name better reflects the way the data structure is organised. Inverted files are data structures that have sequences of documents for every term, in contrast to indexed files that have a sequence of terms for every document.

A standard inverted file structure consists of two main components: a dictionary and a posting list. Figure 2.1 illustrates the organisation of an inverted file structure. Every term in the dictionary has a pointer that indicates the start of a sub-list of postings. This sub-list records all documents where the term can be found. Normally the posting list addresses the documents by a document ID, which requires a third structure to restore document information.

The inverted file structure can be augmented with extra features in the

dictionary and in the posting list. The addition of term context for the retrieval of structured documents such as XML [22] normally represents a modification in the dictionary structure. On the other hand, the addition of the term position, which is used to retrieve phrases, represents modifications in the posting list.

The inverted file structure can be enhanced with the use of compression techniques, which reduce the size of the index and therefore memory usage. Compression techniques are able to improve the overall IR speed performance because they enable the encode, decode, read and write of compressed data to be faster than the read and write of uncompressed data [86, Chapter 5].

Memory usage in IR, including the storage of the dictionary and posting list, can greatly affect the speed performance; therefore, minimising the memory requirements is paramount for IR systems. One method to reduce these memory requirements is to compress the dictionary and posting list. However, compression techniques as shown by Trotman [150] may not always improve system performance, particularly if computationally expensive methods are used.

2.1.1 Retrieval Models

According to Fuhr [42], retrieval models are the theoretical basis of IR, which not only involve how the search is performed, but also how the documents are represented. IR models handle the ordering of results in distinct ways. Some models assume that all documents that satisfy a query are equally relevant; others assume that there is an uncertainty of relevance between documents and queries, which can be used to rank the documents in an order that is believed to place more relevant documents at the top of the list and less relevant ones at the bottom.

The Boolean model refers to the search strategy that retrieves documents that are 'true' for a Boolean query [153, Chapter 5]. Boolean queries contain terms, which define different sets of documents that should be retrieved, and Boolean operations, which define the set of operations that should be applied

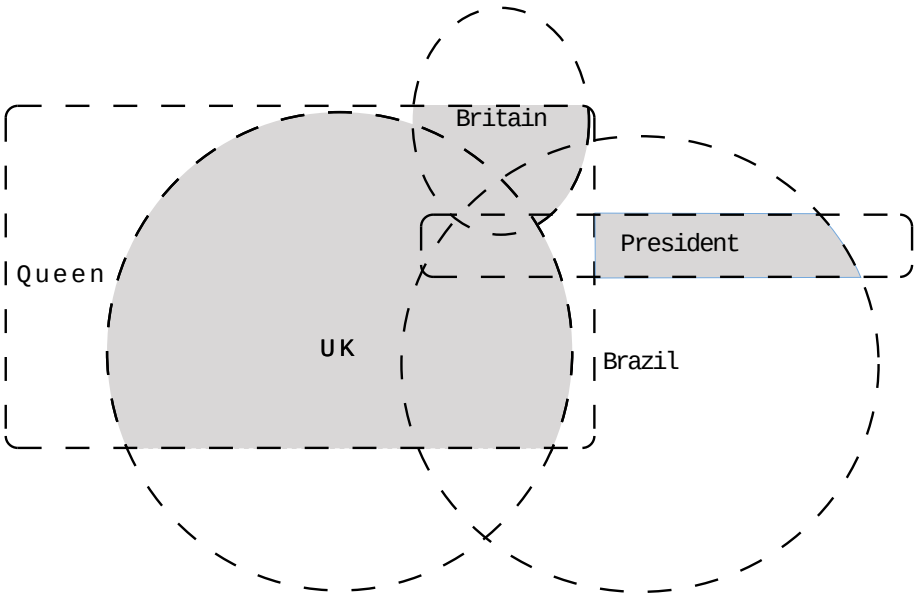


Figure 2.2: Example of retrieval of Boolean query "*(Brazil AND President AND NOT Queen) OR ((Britain OR UK) AND Queen)*". The shading represents the retrieved set.

to these sets in order to generate a final set of documents. Boolean queries are very specific on which terms are used and not used. For instance, Figure 2.2 illustrates the retrieval of the Boolean query "*(Brazil AND President AND NOT Queen) OR ((Britain OR UK) AND Queen)*". The query specifies that the term *Queen* should not appear in the same documents as *Brazil*, unless it appears together with the term *UK* or the term *Britain*.

Probabilistic models try to estimate the probability of relevance of a document according to a user query [153, Chapter 6]. Some probabilistic models such as the binary independence model assume that document terms are independent of each other [86, Page 222], which facilitates the computation of a ranking function. Rijsbergen [153, Chapter 6] states that independence is assumed in probabilistic models only because the calculations otherwise would become intractable. Therefore, the assumption of term independence facilitates the use of simpler corpus statistics such as term frequency and document length, in order to estimate the probability of relevance.

Relevance feedback utilises the feedback from users on the relevance of a given document to improve a second-stage retrieval. The main goal of relevance feedback is to apply the user response to 'move' the query in the direction of relevant items and away from nonrelevant ones [128]. Given the assumption that the first documents in a retrieval set are more likely to be relevant, it is possible to create an automatic relevance feedback mechanism, known as pseudo-relevance feedback [81].

Relevance feedback is associated with most retrieval models, but it is particularly important for probabilistic models since both positive or negative indications of relevant documents can provide more precise statistics about the relevance of terms. Without this information, probabilistic models have to estimate query term probabilities solely based on corpus statistics and must assume an equal chance of unseen terms to contribute towards the relevance of a query.

Language models differ from probabilistic models in that instead of calculating the probability of a document being relevant to a certain query, they calculate the probability of a query being randomly generated by a language model of a certain document. Documents that are more likely to be the language model of a certain query are ranked higher in a retrieval set.

Ponte and Croft [114] were the first to propose the use of language models in IR. They show the feasibility of the method and report improved results over the INQUERY [20] ranking formula, which is a probabilistic approach. More language models have been applied to IR since this work.

Berger and Lafferty [6] describe the process of query formulation by users as a succinct translation of an ideal document. From this idea, they propose a translation model that calculates the chance of a query to be a translation of a document. Every term in the model has a probability of being translated by itself or to another related term. This provides the basis for further work on cross-lingual retrieval such as Xu et al. [169].

Translation models are applied to the QA task by Murdock and Croft [100]. In their work, the use of a translation model in the passage retrieval task

outperforms a query likelihood baseline by allowing matches between related terms and thus increasing the number of matches between passages and queries.

One of the most well known models in IR is the vector space model, of which we will provide a detailed explanation in the next section, and propose some extensions in Chapter 4.

2.1.1.1 Vector Space Model

The vector space model [125] represents documents as vectors with the same number of dimensions as the number of terms in the language. This representation allows the computation of similarity between documents as the angular difference between these vectors. A standard approach for computing similarity is to use the cosine of the angle between two document vectors. Hence, two document vectors will have a zero angular difference (cosine similarity of 100%) if they share the same terms with the same proportional frequencies. In the same way, two document vectors have no cosine similarity if they are represented by perpendicular vectors, i.e. they do not share any terms.

Figure 2.3 illustrates the vector representations of four documents labelled as follows:

- D_{red} ($\rightarrow$) containing the terms {*Brazil*, *Brazil*}
- D_{blue} ($\rightarrow$) containing the terms {*UK*, *Queen*}
- D_{green} ($\rightarrow$) containing the terms {*UK*, *UK*, *Queen*, *Queen*}
- D_{brown} ($\rightarrow$) containing the terms {*Brazil*, *Brazil*, *Queen*}

Because of the angular differences between vectors, we can measure differences in the cosine similarity. For instance, documents D_{green} and D_{blue} contain the same terms with the same relative frequency, which creates two vectors with zero difference in angle between them. The similarity of D_{red} to D_{blue} or D_{green} is zero, since D_{red} does not share any term with D_{blue}

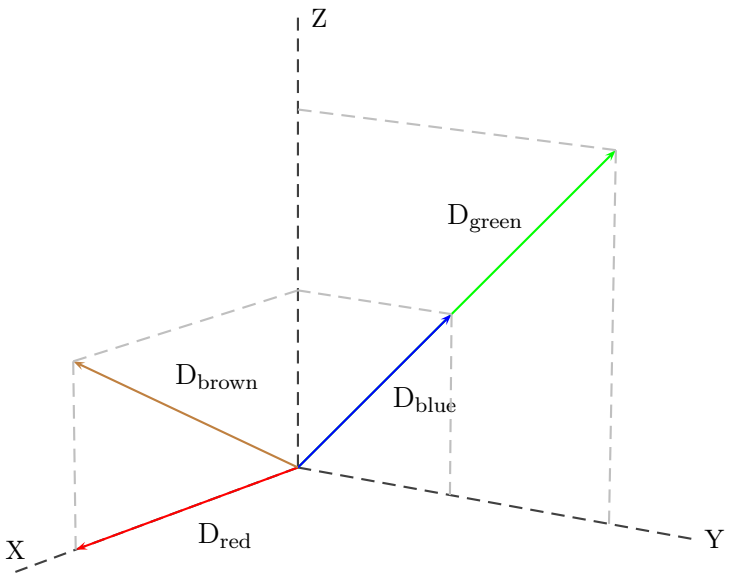


Figure 2.3: Vector representation of documents D_{red}, D_{blue}, D_{green} and D_{brown}. The axes represent the terms as follows: X=*Brazil*, Y=*UK*, Z=*Queen*.

or D_{green}, and therefore they are perpendicular in the vector space. D_{brown} is more similar to D_{red} than the other documents since the angular difference between D_{brown} and D_{red} is smaller (26.6 degrees) than the difference between D_{brown} and D_{blue} or D_{green} (71.6 degrees). This is because D_{brown} shares the only term that D_{red} contains (*Brazil*) with the same frequency, while it shares one out of two terms with D_{blue} and D_{green}.

The vector space model differs from the Boolean model in many ways. The vector space model builds a ranking list of documents ordered by their vector representation similarity to a query. In addition, the vector space retrieval adds documents to its retrieval set according to a logical disjunction model. The only documents that are not included in the retrieval set are those that do not share any terms with the query, i.e. when the document vector is perpendicular to the query vector.

2.1.2 Evaluation

The evaluation of IR systems is strongly bound to the definition of relevance. Relevance is a difficult matter to judge. The task of categorising documents as relevant or irrelevant must be made according to whether they satisfy an information need, which in most cases is not clearly defined by the query. Even specific queries that have clear information needs can have different relevance judgements for its retrieved documents by different users.[1] It is also important to note that the dichotomy of relevance is clearly incorrect because documents may not entirely satisfy the information need at hand; in other words, some documents will have partial relevance.

2.1.2.1 Evaluation Framework

Despite the difficulties in establishing a generic model of relevance, there is a constant effort by researchers to build test collections for IR. Competitions such as the Text REtrieval Conference (TREC) [55], the Cross-Language Evaluation Forum (CLEF) [14], the NII Test Collection for IR Systems (NTCIR) Workshops [67], and the Initiative for the Evaluation of XML retrieval (INEX) [49] have started as early as 1992. These meetings and their several specialised tracks are of extreme importance towards shaping today's IR tools.

The IR evaluation competitions provide a task description and a common evaluation framework that allows systems to be compared against each other. They manually evaluate system results using human assessors, where systems are ranked according to their performance in the task. The judgement evaluations are made publicly available after the competition so that systems and techniques that have not participated in the competition can perform their own evaluations.

The validity of such tests has been disputed due to completeness problems, since they cannot guarantee that an unseen document is not irrelevant.

[1] These issues have been examined by studies on personalised IR, which we will not address in this thesis.

	Retrieved	**Not Retrieved**
Relevant	true positive	false negative
Irrelevant	false positive	true negative

Table 2.1: Document classes used to evaluate IR results

However, the pooling method has been defended in the TREC competitions by Zobel [177] and Voorhees [158] by demonstrating that removing participant systems from the pool and re-evaluating them using the remainder of the pool does not change their final ranking in comparison to other systems.

Another problem with these competitions is that the ambiguous understanding of relevance may cause an assessor to unintentionally penalise or benefit some systems. However, Voorhees [158] argues that there is no evidence that the subjectivity of judgements by different assessors affects the comparative evaluation of systems.

2.1.2.2 Metrics

The standard IR evaluation framework considers documents as either relevant or irrelevant for a certain information need. Thus, under this framework, a perfect IR system would satisfy an information need by returning all of the relevant documents and none that are irrelevant. Table 2.1 shows a document classification used for the evaluation of IR results.

The evaluation of IR techniques focuses on explaining how the retrieval performs according to the classes in Table 2.1. For instance, if a system returns a higher proportion of true positives to false positives, it indicates that the results of this system are mostly relevant. A system may also have a higher proportion of true positive to false negatives, which indicates that most of the relevant information in the document base has been found. The metrics used to provide this information are called precision, recall and F-measure.

Precision is the proportion of retrieved documents that are relevant. Equation (2.1) defines precision as the number of true-positive documents

divided by the number of documents retrieved, regardless of whether they are true positives or false positives.

$$Precision = \frac{|true\ positive|}{|true\ positive + false\ positive|} \tag{2.1}$$

Recall is the proportion of relevant documents that are retrieved. Equation (2.2) defines recall as the number of true-positive documents divided by the number of relevant documents, regardless of whether they are true positives or false negatives.

$$Recall = \frac{|true\ positive|}{|true\ positive + false\ negative|} \tag{2.2}$$

Different IR tasks may require different types of systems. For instance, one task may require only one relevant document, which calls for an IR system that has a high precision; other tasks might need all information regarding a certain subject, therefore requiring IR that focuses on recall. Because the importance of precision and recall may not be always clear, **F-measure** is used as a metric to define whether an improvement is achieved in terms of recall and precision combined. Equation (2.3) defines F-measure as the harmonic mean of precision and recall.

$$F\text{-}Measure = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \tag{2.3}$$

F-measure can also be formulated as shown in Equation (2.4), as the weighted harmonic mean if the importance of precision and recall is known for a certain task. Weights of $\beta > 1$ prioritise recall, $\beta < 1$ prioritise precision, and $\beta = 1$ gives the balanced mean of Equation (2.3). F-measure is commonly referred to as F1 for $\beta = 1$, and F2 for $\beta = 2$.

$$F\text{-}Measure = \frac{(\beta^2 + 1) \times (Precision \times Recall)}{(\beta^2 \times Precision + Recall)} \tag{2.4}$$

When comparing the performance of IR systems, a set of queries are used and the average precision for all these queries is calculated. The average precision measures the precision of an IR system across different recall levels.

Therefore, it gives the expected precision for a certain query and IR system, disregarding the number of documents retrieved. The average precision of a query q is calculated as shown in Equation (2.5).

$$AveragePrecision_q = \frac{1}{|D|} \times \sum_{j=1}^{|D|} \frac{1}{j} \times (\text{number of true positives in } \{d_1, \ldots d_j\}) \tag{2.5}$$

Where D is the set of documents retrieved for query q.

The **Mean Average Precision (MAP)** of an IR system is calculated using a complete set of evaluation queries Q, as shown in Equation (2.6).

$$MAP = \frac{1}{|Q|} \times \sum_{i=1}^{|Q|} AveragePrecision_{q_i} \tag{2.6}$$

2.2 Question Answering

According to Simmons [136], the field of QA has a history dating back to 1959. In his 1965 survey, Simmons found more than a dozen QA systems. Even with this quantity of work, Simmons observes that the results were promising but not practical: most of these early QA systems worked over databases and were very restricted in domain due their limitation in memory and processing power.

Today's QA systems focus more on answering questions by searching natural language texts, rather than information in structured databases. This change in focus is likely due to the large amount of texts that are electronically available, and due to the improvements in the research areas of computational linguistics and IR.

Many recent QA systems participate in the QA Tracks of various conferences. Perhaps the most influential conference is hosted by the National Institute of Standards and Technology (NIST), initially at TREC and now at the Text Analysis Conference (TAC). There are other QA evaluation forums such as the QA track at CLEF, and the QA and the Cross-Language

QA (CLQA) tracks at the NTCIR Workshops. The QA tracks at CLEF focus on non-English monolingual QA and cross-language QA for European languages, while the NTCIR originated as a monolingual Japanese QA track to now become a cross-language QA of Chinese, Japanese and English. These QA tracks provide important common environments where new techniques are shown and systems are tested. However, because this thesis studies the IR for QA tasks in an English monolingual environment, we will only focus on the QA tracks at NIST.

2.2.1 Question Answering Tracks at NIST

Since 1992, the TREC conference has been supported by NIST with the following goals [160]:

- to encourage research in IR based on large text collections;
- to increase communication among industry, academia and government by creating an open forum for the exchange of research ideas;
- to speed the transfer of technology from research labs into commercial products by demonstrating substantial improvements in retrieval methodologies and real-world problems; and
- to increase the availability of appropriate evaluation techniques for use by industry and academia, including development of new evaluation techniques more applicable to current systems.

From 1999 (TREC-8) to 2007 (TREC 2007)[2], a QA track was included in order to instigate competitive research in the QA field. In 2008, TREC's QA track was moved to TAC, which is also hosted by NIST. TAC was created to promote research in NLP by providing large test collections, common evaluation procedures, and an environment that promotes the sharing of results.[3]

[2]Since 2001, TREC has been identified by the year in which it is held.

[3]See `http://www.nist.gov/tac`

Over the years, the QA task at the NIST conferences has changed considerably. The first QA track (TREC-8) contained a set of 200 fact-based questions and utilised a 1.9 GB corpus [154, 163]. The questions were selected from a large pool of questions proposed by the task participants and the NIST TREC team, and from the log files of FAQFinder [18], which is a QA system that uses frequently asked questions as its knowledge base. In this first QA track, for every question there was at least one answer in the corpus. The QA systems needed to provide five answers at most, with references to the documents from the TREC corpus on which the answer was based.

TREC-9 used a set of 693 fact-based questions that had to be answered based on a 3 GB corpus [155]. Questions were real-world questions, extracted from Microsoft Encarta and Excite logs. Changes in the question types could also be found in this track. For example, Elworthy [36] observes that *who* questions in TREC-8 only contained questions asking for an entity name, while TREC-9 contained questions that required a description as its answer. The difference is shown in the following examples cited by Elworthy [36]:

- "*Who is the richest person in the world?*" (entity)
- "*Who is Desmond Tutu?*" (description)

For the TREC 2001 QA main track the systems had to answer 500 open-domain fact-based questions, this time from MSNSearch and AskJeeves logs, and rely on the same 3 GB corpus of TREC-9, which contains nearly one million news articles [156]. This track differed from the previous one because there were no guarantees that an answer would be found in the collection. This introduced a challenge for competitors since they had to create confidence scores to define their certainty of having found an answer to a question or not.

The TREC 2001 QA track also introduced the List task, where QA systems had to answer a list-type question. These questions required to receive a certain number of answers to be right. For example, the question "*Name*

Topic: Franz Kafka

Questions:

- Where was Franz Kafka born?
- When was he born?
- What is his ethnic background?
- What books did he author?
- Other

Figure 2.4: Example of a topic-centred question used in the QA track of TREC from its 2004 edition

two capital cities in Europe" required an answer containing two names of cities. Any answer containing only one capital city in Europe was not entirely correct.

According to Voorhees [157], the main difference in the QA track of TREC 2002 from the previous year was that the QA systems had to return the exact answer, not just a part of text that contained the answer.

The TREC 2003 main task in the QA track introduced three types of questions: factoid, list and definition questions [159], which were originally extracted from Microsoft and AOL[4] logs. The TREC 2003 QA track also featured a passage task, where the answer was presented using a 250 character span of text. The passage task was discontinued in subsequent years.

The main modification of TREC 2004 QA track from the previous years concerned the set of questions, which were grouped into different topics [161]. Every topic had a few questions associated to it, and ended with an *other*-type question. The *other*-type questions required the systems to present relevant information about the topic that had not been covered by previous questions. Figure 2.2.1 gives an example of a set of questions regarding one specific topic used in TREC 2004 QA track.

TREC 2005 did not present many differences from TREC 2004; the most significant changes included the inclusion of events as topics, and the possibility to submit the list of documents retrieved as an effort to improve the

[4] `http://www.aol.com`

IR stages of QA systems [162]. This was the first effort from a QA track to provide a means of evaluating the IR stage for the QA task.

The TREC 2006 main QA track was essentially the same as TREC 2005. The main difference involved time sensitive questions, which were previously judged according to the time frame from the supporting document [32]. If a question was presented in the present tense, the correct answer would need to be based on the latest information available for that question. For instance if a question asked for the current president of a certain country, and different documents present contradictory information, only the latest information would be considered correct.

The TREC 2007 main QA task maintained the format from previous QA tracks, but the corpus used had changed considerably [30]. For that year, the AQUAINT-2 [5] corpus was used together with the Blog06 [82]. The AQUAINT-2 corpus is similar to the AQUAINT [50] corpus because it comprises of a collection of newswire texts. The Blog06 corpus, on the other hand, contains a collection of raw HTML that had been copied from WWW sources from late 2005 to early 2006.

TAC-2008 was a continuation of the main QA task of TREC 2007 with considerable changes.[6] Factoid questions were no longer used and the task consisted of answering opinion-type questions regarding the documents from Blog06 and AQUAINT-2.

Even though the systems' results are substantially different for the QA tracks, the approach to the task mostly followed a standard system architecture (Figure 1.1 on page 18). For instance, Voorhees [161] writes:

> "*Systems generally determine the expected answer type of the question, retrieve documents or passages likely to contain answers to the question using important question words and related terms as the query, and then perform a match between the question words and retrieved passages to extract the answer.*"

[5] `http://www.nist.gov/tac/data/data_desc.html`

[6] `http://www.nist.gov/tac/tracks/2008/qa`

2.2.2 Evaluation Metrics

Perhaps the most common metric used to evaluated QA system is the **Mean Reciprocal Rank (MRR)**. MRR is defined in Equation (2.7) as the average of the reciprocal rank, which is the inverse of the rank of the first correct answer, for every question in the evaluation set:

$$MRR = \frac{1}{|Q|} \times \sum_{i=1}^{|Q|} \frac{1}{rank_{q_i}} \tag{2.7}$$

Where Q is a set of questions, and $rank_{q_i}$ is the rank of the first correct answer of question q_i.

MRR is normally used for systems that return a ranked list of answers. MRR rewards systems that return answers near the top of the list, and penalises systems that return answers in lower ranks. The difference between MRR and precision is that precision requires a fixed amount of answers to be calculated and when this amount is defined the rankings of the correct answers are irrelevant. For instance, consider the following two ordered lists of answers, where denotes a correct answer and denotes a wrong answer:

$$RA_1 = \{ \qquad \}, RA_2 = \{ \qquad \}$$

In order to evaluate the precision of these lists of answers, we ignore the position where the correct answers appears and both lists are considered equally good with a precision of 20% (1 correct out of 5). On the other hand, for the reciprocal rank, the number of total answers returned is irrelevant, and only the first correct answer is measured. For the above two answer lists, the reciprocal rank values are 1.0 for RA_1 and 0.25 for RA_2, demonstrating the usefulness of the metric for this QA setup. MRR reflects the average position of the answer for a given system, which can be explicitly calculated by inverting the MRR value ($\frac{1}{MRR}$).

Even though MRR is regarded as a better metric than precision to evaluate a QA system that returns a list of possible answers [116], precision is still important when multiple answers are allowed. In the QA track

of TREC 2003 [159], when list questions (questions that require a list of items as answers) were introduced, the metric used for evaluation was the F-measure with equal weights assigned to recall and precision; the definition type of questions (other-type from 2004 onwards) were also evaluated using F-measure but with different weights (with more importance on recall).

The MRR metric is used for the evaluation of most QA competitions, such as TREC [164], CLEF [84] and NTCIR [46], when more than one answer is allowed. However, MRR is dropped when only one answer is required for every question. TREC has adopted the single answer requirement from TREC 2002 QA Track [157]. The QA track at CLEF permits from time to time more than one answer per question, which allow systems to be evaluated using MRR [41, 83]. For the competitions with a single answer setup, the evaluation is done in terms of accuracy. **Accuracy** is measured as the proportion of correct answers to the total number of questions (Equation 2.8):

$$Accuracy = \frac{1}{|Q|} \times \sum_{i=1}^{|Q|} Correct(q_i) \tag{2.8}$$

Where Q is a set of questions, and $Correct(q_j)$ is equal to 1 if question q_j is correctly answered, otherwise $Correct(q_j)$ is equal to 0.

QA competitions have also experimented with different evaluation scores such as K1 measure, confident weighted score and nugget pyramids [31, 32, 47].

2.2.3 Question Answering Framework

Several systems in the literature answer questions based on text corpus information. Some of these systems operate only for domain-specific texts while others are open domain and work with generic texts. While domain-specific QA systems apply extensive NLP techniques and domain-specific resources, open-context QA systems are limited in NLP resources and often require to make a trade-off between QA performance and processing time. For instance, the MULDER QA system [72] modifies the requirements of its parser

in order to obtain better or quicker results.

Most QA systems reduce the complexity of dealing with large amounts of text data by using IR techniques. Roberts and Gaizauskas [119] affirm that this may be because NLP researchers have not spent time optimising the access of their data structures and their algorithms.

Even today's web-based QA systems tend to reduce their systems complexity by using known WWW search engines to perform the IR and Passage Retrieval tasks. This makes the process of web-based QA very similar to other corpus-based methods, where perhaps the only difference is the size of the corpus used.

In regards to IR, most QA systems present no major differences even when they use different types of corpora. This was observed by Voorhees [161] where most QA systems in TREC 2004 follow the traditional pipeline architecture shown in Figure 1.1 (page 18), with the standard stages of a QA system being Question Analysis, IR and Passage Retrieval, and Answer Extraction.

2.2.3.1 Question Analysis

The primary task that allows a QA system to answer a natural language question is the interpretation of the information contained in the question. According to Moldovan et al. [94], when the question analysis fails, it is hard or almost impossible for a QA system to perform its task. The importance of the question analysis is very clear in the QA system of Moldovan et al. since this task is performed by 5 of the 10 modules that are part of their system.

One simple approach for analysing questions is to divide the task into two parts: finding the Expected Answer Type (EAT), and finding the question focus.

For the task of finding the question focus, the simplest approach is to discard every stopword of the question and to consider the remaining terms as the focus representation.

Many systems use a set of handcrafted rules for finding the EAT [1, 24, 59, 74, 95]. This is normally sufficient for a large amount of questions because, according to Lin [74], natural language questions obey Zipf's law [176]. Lin [74] argues that just a few question schemas are able to handle a large number of questions. Normally these schemas are written as regular expressions, and the task of finding the EAT is a matching problem between questions and regular expressions. Every regular expression will have an associated EAT that will be assigned to a question if it matches its pattern. Question patterns are also used to find the question focus. This approach can be seen in the work of Harabagiu et al. [54], where they apply question patterns to find the focus of definition questions.

According to Chen et al. [24] the EAT recognition falls into two broad groups: those based on lexical categories and those based on answer patterns. EAT analysis based on lexical categories identifies some lexical information present in the questions, while analysis based on answer patterns predicts the EAT by the recognition of certain question types.

Hovy et al. [59] build a QA typology in order to produce a range of specific to general EAT. Question patterns are assigned for every answer category, and for these some examples of questions are provided. In a later work, Hermjakob [56] describes their intentions of migrating from manually defined rules to automatically learned ones, but there is no further evidence that this work ever took place.

Li and Roth [73] created a dataset of 5,500 factoid questions annotated with the EAT. They use a machine-learning approach and achieved over 90% accuracy in the automatic classification. Similar results have been found by Zhang and Lee [174] when using a system based on support vector machine trained over the Li and Roth [73] dataset. We also demonstrated in [110, 171, 172] that question classification can be performed reasonably well using simple machine-learning techniques such as trie-based classification and alignment-based learning. The trie-based classifier is detailed in Chapter 3.

The Li and Roth [73] dataset was augmented with the TREC 2001-2006 questions and some new question types by Shen et al. [135], who use a naïve Bayes classifier to find the EAT of questions.

Paşca and Harabagiu [106] note that question stems like *who*, *where* and *what* can be highly ambiguous, and that some systems implement rules for establishing a relation between question stems and the entity type of the answer. The QA system from Na et al. [101] uses question patterns supported by WordNet [38] to define the EAT of a query. The experiments from Paşca and Harabagiu [106] show that the use of a large semantic database can help to achieve high quality precision over ambiguous question stems for finding the question's EAT.

Moldovan et al [92] describe a hybrid approach to EAT classification that combines precise heuristics and machine-learning algorithms. A hybrid approach using manually encoded rules and an automatic classifier based on lexical, syntactic and semantic features is also used by Schlaefer et al. [131].

2.2.3.2 Information Retrieval and Passage Retrieval

In most QA systems, the IR stage follows the question analysis stage. IR reduces the corpus to a smaller number of documents that are likely to contain the answer to a question. In QA, this is necessary because the answer extraction phase is normally the most costly phase and reducing the amount of information that needs to be processed is paramount. Because the answer extraction component only considers the documents given by the IR system, errors in the retrieval phase can seriously compromise the final QA results. Thus IR is a critical process of the QA pipeline.

Despite the importance of the IR phase, most QA systems regard IR tools as black boxes, such that most work is performed on either the input (query) or the output (documents). The following list shows some IR tools and QA systems that use them.

- IR tools built using the *Lemur Toolkit*[7], such as the *Indri Search En-*

[7] `http://www.lemurproject.org`

gine [138] are used by *Alyssa* of Shen et al. [135], *Ephyra* of Schlaefer et al. [131], *Pronto* of Bos et al. [11].

- The *Lucene Search Engine Library*[8] is used by Katz et al. [68], Greenwood [51], Razmara et al. [118], *QuaLiM* of Kaisser [64], Saxena et al. [129], Neumann and Sacaleanu [103].

- The *Xapian Search Engine Library*[9] is used by de Pablo-Sánchez et al. [33], Whittaker et al. [166], *AnswerFinder (AF)* of van Zaanen et al. [170], Tomás et al. [149].

- *Okapi* [122] is used by Roberts and Gaizauskas [119], and Elworthy [36].

- A modified version of *SMART* [126] is used by Abney et al. [1].

- *Web Search Engines* (Yahoo!, Google, etc) are used by: *Ephyra* of Schlaefer et al. [131], Tomás et al. [149], Aranea of Lin [76], Brill et al. [16], besides other specific web QA tools such as *MetaQA* of Pizzato and Mollá [111], *START* of Katz et al. [69], *AnswerBus* of Zheng [175], *LAMP* of Zhang and Lee [173], *MULDER* of Kwok et al. [72].

Lucene is also used in the work of Tiedemann [146], who explores this search tool in a manner beyond a black box. In his work, different layers of linguistic information are added to the IR tool, which are later combined to extract text passages with the assistance of weights defined by a genetic algorithm. This work will be further discussed in Section 2.4.1.1.

Because QA requires answers that consist of only a few words, reducing the size of the retrieved document set even further is common practice in QA systems. This reduction normally takes place as passage retrieval, which is the task of finding the document parts that will answer the questions at hand.

[8] `http://lucene.apache.org`
[9] `http://www.xapian.org`

Some comparative research has been carried out with passage retrieval and QA. Clarke and Terra [26] point out that full document retrieval may be desirable for QA since it yields higher precision. However, Clarke and Terra [26] have not compared the impact of document retrieval and passage retrieval on a working QA system, which could suggest contradictory results since documents are likely to be noisier than passages in regards to finding an answer to a question. Roberts and Gaizauskas [119] evaluate different approaches of passage retrieval for QA. In their study, better results are obtained when the passage retrieval engine retrieves a full document and further breaks the document into passages.

Monz [97] defined the minimal span weighting scheme that gives preference to small passages that contain a large number of query words. Improvements over standard passage retrieval methods are also achieved by Usunier et al. [151]. The authors apply the RankBoost machine-learning algorithm over the results of a conventional passage retrieval engine, improving the rank of passages that might contain answers to questions.

Moldovan et al. [94] compare the Boolean model and the vector space model for document retrieval in a QA system, and have found the Boolean model to be more effective. Moldovan et al. [94] argue that the occurrence of many query terms is a better relevance indicator than the multiple occurrences of a single query term.

The system of Paşca and Harabagiu [106] employs a Boolean IR engine for finding paragraphs that may contain the answer. In order to have a certain required number of paragraphs, keywords are dropped from the search engine query when the search is too restrictive and others keywords are added to the query when too many paragraphs are found.

2.2.3.3 Answer Extraction

In the QA task, after finding the best documents for answering a question, it is important to identify its potential answer. The answer extraction component of a QA system focuses on pinpointing the location of a probable

answer for a certain question. Some earlier QA systems and some web QA systems such as AnswerBus[10] developed by Zheng [175] and QuaLiM[11] of Kaisser [64] focus on providing a snippet of text that can answer a question. Even though an exact answer string may not always be the optimal answer from the user's point of view [65], most current systems focus on delivering short and exact answers. This is likely to be inspired by the stronger constraints set in the recent QA competitions (see Section 2.2.1).

The answer extraction phase in QA is where most distinctions among systems are found. Nevertheless, most QA systems in NIST competitions include some kind of named-entity component in their systems (e.g. [94, 95, 173]). This is perhaps due to the large amount of factoid and list questions, which normally require a named entity as the answer. Razmara et al. [118] uses a modification of Aranea [76], which includes the use of a named-entity recogniser and gazetteers to identify known named entities. Saxena et al. [129] uses a standard approach to answer extraction, which includes the use of named-entity recognition and surface pattern matching. They also use evidence from WordNet semantic classes and from Wikipedia[12] to rank possible answers.

Besides named-entity recognition, other NLP techniques are also used in the answer extraction phase of the QA process. For instance, part-of-speech and grammatical relations are also used in the system of Buchholz [17], where if a similar question representation is found in the corpus, the chunk of text that matches a special marker is considered as a possible answer. Other systems that perform syntactic dependency matching include Cui et al. [28], and Shen and Klakow [133]. The logic prover COGEX [91] is the decisive tool used by Moldovan et al. [93] when deciding whether an answer candidate will be selected as the final answer for a question.

Some of the top-scoring systems in the latest TREC competitions have used some type of semantic matching of relations between predicates and

[10] `http://www.answerbus.com`

[11] `http://demos.inf.ed.ac.uk:8080/qualim`

[12] `http://www.wikipedia.org`

arguments. Schlaefer et al. [131] uses ASSERT [115] as their Semantic Role Labelling (SRL) tool for their answer extraction process. However, because SRL is a time-intensive task, they apply a series of heuristics, which includes the presence of named entity of the EAT (if known), to decide on the sentences that are going to be processed. This process re-ranks passages according to their predicate similarity, which is calculated using the similarity of verbs and arguments via an extended version of the Jaccard coefficient [61]. After the re-ranking is done, Schlaefer et al. [131] extract the answer based on the presence of the named-entity class from the EAT (if known), or on the existence of an argument in the answer string whose role is missing in the question representation.

Shen et al. [135] view answer extraction as a graph-matching problem where candidate answers are ranked by the similarity between the semantic structures of the candidate answer and questions.

Another QA system that makes use of SRL is the CHAUCER-2 system by the Language Computer Corporation [57]. They use an extensive combination of resources and techniques, including a specialised IR component.

2.2.4 Question Answering and Linguistic Resources

The importance of NLP techniques in QA is demonstrated by Moldovan et al. [94], who show that if no NLP techniques are used in the answer-extraction process, a very low precision (0.028) is obtained. By incrementally adding NLP modules, the precision raises gradually and substantially (up to 0.572). According to Moldovan et al. [94], the overall performance of a QA system is directly related to the depth of the NLP resources. They also state that today's QA systems perform better when relevant passages and candidate answers are clearly defined.

Some systems rely on linguistics resources such as WordNet to improve the QA task. Harabagiu and Moldovan [53] uses WordNet and an online dictionary as their system knowledge base. In further work by Paşca and Harabagiu [106] and Harabagiu et al. [54], WordNet is used to assign the

EAT category. Their system matches questions' keywords with WordNet synsets, and by finding dependencies between synsets, derives an EAT from it. WordNet is also used to support manually defined patters for answer extraction in Na et al. [101].

WordNet, despite its large size and wide use, only contains a small percentage of proper nouns. According to Mann [85], because there is no ontology for proper nouns with similar coverage as nouns in WordNet, most QA systems rely on named-entity recognisers to identify proper nouns. Mann [85] presents a limited ontology of proper nouns, built from approximately 1 GB of news texts. The ontology is constructed by searching for instances of a common noun followed by a proper noun. Mann [85] separates each description and its proper noun in its own synset. This approach, even with the drawback of not considering polysemous words and complex noun constructions, has obtained several times more named entities on specific ontology branches (like lead singers, folk singers, or even singers in general) than WordNet has. Mann [85] is able to improve recall and to slightly decrease precision of QA using the proposed ontology.

Perhaps the most common use of the WordNet, and other semantic databases, is through a technique called query expansion. This technique, according to Strzalkowski et al. [140], modifies an IR query to better resemble the documents it is expected to retrieve. This can be seen in the work of Kwok et al. [72] where WordNet is used to modify the question in order to obtain the right keywords to be used in a search engine for a certain question. However, the use of such resources may not necessarily reduce the terminology gap between question and documents as shown by van der Plas and Tiedemann [152].

Some NLP tools, such as grammatical parsers and Part-of-Speech (POS) taggers, are broadly used in all QA applications. Shapaqa [17] determines grammatical relations in the questions and tries to match them with grammatical relations (with the same keywords) in the documents. Hovy et al. [59] presented a system at the TREC-9 QA track that matches parse trees

from document's segments and questions. According to the authors, the challenge of this approach is to perform the task fast enough to be practical in large collections of text. The QA system developed by Mollá [95] matches questions and answers by measuring the overlap of words, grammatical relations and minimal logical forms of questions and document's sentences. In his study, the Connexor Functional Dependency Grammar Parser from Connexor Oy[13] is used along with the LaSIE named-entity recogniser of the GATE system[14].

Scott and Gaizauskas [132] also use the LaSIE system (by the GATE interface) for NLP. For the text analysis, the following modules are used: Tokeniser; Gazetteer Lookup; Sentence Splitter; Brill Tagger [15]; Tagged Morph; buChart Parser [60]; Name Matcher; and Discourse Interpreter. [15]

The QA system developed by Elworthy [36] matches questions with answers by extracting the document syntactic nodes that are associated with the logical forms of the question, and removing the extra words of the answer when they do not fit into the TREC-9 format.

2.3 Information Retrieval for Question Answering

Most of today's QA systems access the WWW only by way of an IR engine. According to Roberts and Gaizauskas [119], the reason for using IR engines in QA tasks is that IR is far less time consuming than most NLP techniques. In this section, we will examine work that looks deeper into the IR problem and develops custom solutions for their IR modules. We will first look into offline QA, which is the task of finding answers to questions while processing the corpus for the QA task. Similar to the IR task, answers to pre-defined question patterns are stored using an efficient data structure so that they can be quickly retrieved when the corresponding question is asked.

[13] `http://www.connexor.com`

[14] `http://gate.ac.uk`

[15] Projects seeking to use GATE will find that the information extraction module LaSIE was replaced by ANNIE and that the buChart parser is called SUPPLE parser from Gate version 3.1.

2.3.1 Information Retrieval as Offline Question Answering

According to Fleischman et al. [40], IR techniques are too slow when considering the exactness required for QA systems. The authors are referring to the access time of supporting information from external sources, such as Wikipedia and Google; this problem is also found by Kaisser [64]. As an alternative to IR, Fleischman et al. [40] propose the use of a custom built large repository of information to quickly and precisely answer some specific question types. To build such representations, Fleischman et al. use a machine-learning classifier based on manually defined features, and obtain improvements over the state-of-the-art in regards to *who* questions.

Jijkoun et al. [62] address the low recall problem of surface pattern methods of offline QA such as Fleischman et al. [40], by using dependency parsing to extract information between entities that are not necessary adjacent in the surface level. In order to build look-up tables containing information about named-entities, Bouma et al [12] combine anaphora resolution with syntactic relations to expand the coverage of the relations extracted.

In web QA, Lin [74] calls the construction of databases as the federated approach to QA; one such systems is START[16], which is the first known QA system available on the web. According to Katz et al. [69], the START system translates natural language questions into structured queries that are submitted to a common interface to multiple web knowledge bases. This common interface is denominated as Omnibase. The START system transforms questions into object-property-type queries to be used over a specialised WWW database. For example, if a question is asked about a movie, a query is sent to the IMDb web site[17]. The answer for the movie question is then extracted from the already known IMDb format. According to Lin [74], the system is capable of answering 27% of TREC-9 and 47% of TREC 2001 questions using 10 WWW data sources.

In further work, Katz et al. [70] combined web-based and corpus-based

[16] `http://start.csail.mit.edu`

[17] `http://www.imdb.com`

techniques for their participation in the TREC 2003 QA track. Factoid questions are answered by a similar method described by Katz et al. [69], while list questions employ a traditional QA pipeline architecture. In Katz et al. [70], the AQUAINT corpus is processed offline for answering definition questions. For every article, pattern-matching techniques extract information nuggets that fill a relational database. Once the database is completed, the process of answering definition type questions is performed by a database lookup.

The offline creation of fact databases can be very precise for specific and pre-defined question types. However, in order to cover a higher range of question types, QA still needs to incorporate IR processes. As mentioned in Section 2.2.3.2, many QA systems use off-the-shelf IR tools as a black box, and most IR modifications are either on the input as query modifications, or on the output as passage retrieval.

2.3.2 Query Modification

Radev et al. [117] assume the existence of one single IR query that will retrieve the best set of documents containing the answer for a certain question. In their study, in order to find the best query for a question, they present an algorithm based on expectation maximization that gradually performs a set of transformations on the original question.

Most QA systems rely heavily on IR engines; when the documents returned by these engines do not entirely fit the requirements of the systems, query expansion techniques are applied in order to relax or constrain the IR query. Paşca and Harabagiu [106] use query expansion based on lexical, semantic and morphological alterations for generating a wider retrieval set when the initial set of results does not provide the required number of paragraphs.

The work of Bilotti at al. [9] examines whether stemming or query expansion would lead to better results in QA. Their results indicate that stemming gives lower recall, while query expansion using morphological variations of

the query words gives higher recall in comparison to a non-stemming baseline. They also show that adjusting the weights of the expansion terms improves recall even further.

2.3.3 Information Retrieval Models for Question Answering

A comparison between Boolean and vector space retrieval methods were carried out by Moldovan et al. [94]. The authors show that vector space methods are not necessarily better than the Boolean model, in particular when the vector space model prioritise documents containing several instances of the same term over few instances of different terms.

The conclusion that Boolean models perform well for the QA task is also made by Tellex et al. [144]. They perform a comparative evaluation of passage retrieval models, which shows that density-based measures are important for QA. Density-based scoring means that passages that contain terms closer together will rank higher than a passage where terms are far apart.

Simple models of retrieval can also perform well for QA. Chang et al. [23] compare different retrieval methods such as passage size granularity and they decide on retrieving passages of the size of single sentences by splitting documents into sentences and indexing these sentences separately using a vector space model. Evidence for the effectiveness of single sentence retrieval is also found by White and Sutcliffe [165], who state that nearly 90% of the answers of their test questions are supported by single sentences.

Monz [98] defines a retrieval ranking model called minimal span weighting that considers the distance between terms in a document, where this information is used to vary the passage size returned to the QA system. Documents that contain small passages with most query words are more relevant than documents where the query words appear distant to each other. Despite having the assumption that the answers will appear within the boundaries of the passage that contain the question words (which is not necessarily always the case), it shows slight improvements over the standard Okapi [120]

approach for passage retrieval.

Murdock and Croft [100] apply a translation model for the IR of a QA system. The translation model calculates the probability of a question to be generated by a translation of a certain language model, represented by documents. Models that are more likely to be the source model of the translated question are ranked higher. Murdock and Croft [100] address the low similarity between questions and documents by finding terms with similar meanings.

2.3.4 Evaluation Metrics

In the context of IR for QA, retrieval sets are evaluated at cut-off points, which are normally specified by the suffix @n in the evaluation metric. This helps to define how many documents the QA system will request from its IR component when processing questions.

Precision is the proportion of the retrieved documents that are relevant to a certain information need. This translates to QA as the percentage of documents, from all the retrieved ones, that can answer a question. Using a collection of documents and question sets, the average precision at point n (p@n) is the average of the precision of all the individual questions, where precision is the total number of relevant documents divided by the total number of retrieved documents. This differs from MAP (Equation 2.6 on page 33) because p@n has a maximum number of documents (n) for every retrieved set, where MAP considers the whole retrieved set.

For every question q_i in a collection of questions Q, a set of documents d_j can be retrieved and a set of relevant documents can be defined. Since we are evaluating at a cutoff point, the size of both sets cannot be larger than this point, defined as n. Precision p@n in this configuration is defined in Equation (2.9):

$$\mathbf{p@n} = \frac{1}{|Q|} \times \sum_{i=1}^{|Q|} \left\{ \frac{|Relevant(q_i) \wedge Retrieved(q_i, n)|}{|Retrieved(q_i, n)|} \right\} \tag{2.9}$$

Where Q is a set of questions, $Relevant(q_j)$ is the set of documents that are relevant for a question q_j, and $Retrieved(q_j, n)$ is the set of documents retrieved for a question q_j with a maximum size of n.

Roberts and Gaizauskas [119] propose coverage and redundancy to evaluate retrieval sets for the QA task as a more meaningful measure than precision and recall. Coverage and redundancy have the advantage of providing a better insight into the usefulness of certain retrieval sets for QA.

Coverage measures the number of questions that can be answered. Unlike precision, which takes into account the number of relevant documents, coverage evaluates the entire retrieval set on a per question basis based on the existence of at least one document that contains a correct answer. This means that a retrieval set containing a single relevant document will be considered just as likely to answer a question as a retrieval set with several relevant documents. This metric is proposed to be used together with redundancy, which gives the missing information about the number of relevant documents retrieved.

Coverage ($c@n$), as defined in Equation (2.10), is the percentage of a question set that can be potentially answered using the top-n documents.

$$\mathbf{c@n} = \frac{1}{|Q|} \times \sum_{i=1}^{|Q|} f(q_i) \tag{2.10}$$

$$\text{where } f(q_i) = \begin{cases} 0, & \text{if } Relevant(q_i) \wedge Retrieved(q_i, n) \equiv \emptyset \\ 1, & \text{otherwise} \end{cases}$$

As defined in Equation (2.11), **redundancy** ($r@n$) is the average number of documents per question that can provide an answer among the top-n documents retrieved.

$$\mathbf{r@n} = \frac{1}{|Q|} \times \sum_{i=1}^{|Q|} \Big\{ |Relevant(q_i) \wedge Retrieved(q_i, n)| \Big\} \tag{2.11}$$

Coverage and redundancy complement each other since the information of each do not provide total insight of the performance of the IR system.

High coverage is desirable since it indicates that the IR system is able to produce at least one relevant document for most question sets. While this is useful and important, high coverage with low redundancy may not be optimal for some QA systems, for instance those that depend on receiving several instances of the same answer.

Redundancy provides important information for evaluating the question set; however, it does not provide more information to what precision already delivers. The difference between the information that precision and redundancy provide is subtle, as illustrated by the following example. Consider that we are evaluating an IR system using p@100 and c@100, and the retrieval set of a single question contains 50 documents, of which all of them are relevant. The precision for the retrieval set of this question is 100%, and the redundancy measure is 50. However, because the number of documents is smaller than the point of measurement (100), redundancy cannot be converted into a precision measure by normalising its value to n. Despite these differences, for cases of small value of n and a large collection of documents using a retrieval model with a default disjunction operation, it is almost certain that p@n is the same as r@n normalised by the value of n.

2.4 Related Work

2.4.1 Linguistically Motivated Retrieval for QA

This thesis proposes the use of linguistic information such as dependency relations and semantic roles in the IR stages of a QA system; such information can help create IR that is better suited for the QA task. Several examples can be drawn to illustrate the usefulness of such methods; however, there has been little published research that has been able to show any real impact on end-to-end QA performance. One exception is the work of Cui et al. [29], which reports improvements between 50% to 138% in MRR and over 95% in p@1 using dependency relations over word density methods for passage retrieval. According to Cui et al., the reason that IR techniques that use de-

pendency relations do not significantly improve the density based models is because they do not allow partial matches of relations; thus failing to identify semantically similar but syntactically different structures. Cui et al. assume that the dependency path between two words in a question is equivalent to a path between the same two words in a sentence that are known to support the answer to the question. Using this mapping of dependency paths, they train two models to allow the mapping of unknown questions to sentences: one model is based on mutual information and the other model is based on expectation maximization. The evaluation shows that the difference of results between the models is small, however both offer large improvements over standard retrieval models.

The next sections highlight two studies in the area of IR with linguistic information. First, we will describe the work of Tiedemann [145, 148], which uses dependency relations, then we will describe the work of Bilotti et al. [7, 10], which uses semantic roles.

2.4.1.1 Dependency Relations at the University of Groningen

Tiedemann (University of Groningen) proposes a strategy to include linguistic information in IR, which has shown to be beneficial to the QA task [145, 148]. Tiedemann [145] uses several index layers to encode different types of linguistic information; the Lucene search engine to perform the IR task, and the Alpino parser [13] to extract linguistic information, which includes dependency relations, parts of speech, named entities, and the root form of words. This information forms the basis of several index layers, or features [148]. The set of layers used by the author and some examples of the layer contents are shown in Figure 2.5 (excerpt from [12, page 28]). Because each of these layers can be used independently, with different weights of importance, the number of different system configurations is extremely large and an automatic process to find the best parameters is required. Tiedemann uses a genetic algorithm to find an optimal set of parameters that improve the IR for QA with respect to MRR.

layers for each word in each paragraph		
text	plain text tokens	Het embargo tegen Irak werd ingesteld na de inval in Koeweit in 1990
root	linguistic root forms	het embargo tegen Irak word stel in na de inval in Koeweit in 1990
RootPOS	root + POS tag	het/det embargo/noun tegen/prep Irak/name word/verb stel_in/verb na/prep de/det inval/noun in/prep Koeweit/name in/prep 1990/noun
RootRel	root + relation (to its head)	het/det embargo/su tegen/mod Irak/obj1 word/ stel_in/vc na/mod de/det inval/obj1 in/mod Koeweit/obj1 in/mod 1990/obj1
RootHead	root (dependent) + root (head)	het/embargo embargo/word tegen/embargo Irak/tegen word/ stel_in/word na/stel_in de/inval inval/na in/inval Koeweit/in in/inval 1990/in
RootRelHead	dependent + relation + head	het/det/embargo embargo/su/word tegen/mod/embargo Irak/obj1/tegen word// stel_in/vc/word na/mod/stel_in de/det/inval inval/obj1/na in/mod/inval Koeweit/obj1/in in/mod/inval 1990/obj1/in
layers for selected words in each paragraph		
compound	compounds	stel_in
ne	named entities	Irak Koeweit
neLOC	location names	Irak Koeweit
nePER	person names	
neORG	organisation names	
neTypes	labels of named entities	LOC LOC YEAR

Figure 2.5: Layer types used by Tiedemann (excerpt from [12, page 28])

In [148], Tiedemann experiments with different genetic algorithms. These algorithms have shown different run time performances but produce query settings with similar improvements. The genetic algorithms applied in this work find distinct query configurations, which demonstrates that the genetic algorithms converge to a local maximum instead of a global one. Tiedemann observes that this is partially due to the low impact that changes in the query parameters have in the actual retrieval performance.

The author uses MRR as a fitness function for the genetic algorithms, which may be another reason for the difficulty to measure improvements.

Even though MRR is an important metric in QA evaluation, it evaluates a full retrieval set only on the basis of the first correct retrieved item; this may not reflect the true usefulness of the retrieval set, in particular for QA systems that rely on answer redundancy. Therefore, the local maximums obtained in Tiedemann's experiments might optimise the retrieval for most questions in terms of MRR, but it does not necessarily maximise the number of answer-bearing sentences in the retrieval set.

Tiedemann [145] notes that Lucene is used as a black box, because he does not have knowledge about Lucene's internal scoring mechanism and how the output is influenced by changes in the input parameters. However, it is understood that Lucene allows the definition of certain features as more or less important and as required or not. In [145], the best combination of parameters is found to be as the required presence of the root form of words and the root form with part of speech in case of nouns, and high weights for the original words, for the presence of an entity of the EAT, and for the presence of the word in a subject relation. Further experiments in [148] show that several query configurations are found, but no particular query configurations are discussed. Bouma et al. [12] stress that improvements in the retrieval component are not solely based on single word information, but many features that are given high weights refer to the use of dependency information.

Tiedemann has shown that when given the right parameters, the inclusion of linguistic information can produce a retrieval set that closer resembles the QA task than a baseline bag-of-words retrieval. The improvements due to the linguistic information from Alpino are expected to be carried out to the QA system, in particular in the case of Joost[18], which makes heavy use of Alpino in all its modules. However, there is no indication of the impact of the improved IR in the final results of the QA system.

In this thesis, we report the measure of the effectiveness of using linguistic

[18]The QA system developed at Groningen University, where Tiedemann's IR techniques are applied [12].

information, including dependency relations in a unweighted setup. The importance of terms, their lexical, syntactic and semantic information is defined simply by corpus statistics.

2.4.1.2 Semantic Roles at Carnegie Mellon University

Bilotti et al. from Carnegie Mellon University describe a method to retrieve a high precision set of sentences based on the agreement between the verb predicate-argument structures of the questions and their supporting sentences [7, 10]. In Bilotti et al. [10], the authors detail a high recall IR model that is a better fit for the QA task than the bag-of-words model. In Bilotti and Nyberg [7], the authors evaluate this high precision IR technique by itself and using a QA system.

In [10], the authors report that high recall is desirable for QA, while in [7] the main objective is precision. The reason for this change of focus is not made clear. However, one of the claims of [10] is that better recall in retrieval sets can yield improvements in efficiency of QA systems, which is demonstrated by assuming that QA systems have a desired recall value and measuring the number of sentences that need to be processed at that level. Despite this claim, the setup for the QA experiments in [7] uses the same number of sentences for both the bag-of-words retrieval and the verb predicate-argument structure retrieval. Because the number of used sentences is fixed, precision better reflects what is being represented; this is particularly true in cases when the total number of relevant sentences is not known.

Bilotti et al. [10] demonstrate that the use of PropBank-style semantic roles [107] can improve the sentence retrieval in terms of recall and precision for the QA task. A similar evaluation presented in [7] yields similar results; however, an evaluation using the enhanced retrieval set in a working QA system (OpenEphyra[19] from Schlaefer et al. [131]) fails to show improvements over the bag-of-words baseline.

[19] Available at `http://www.ephyra.info`

	MAP	**Accuracy**	**MRR**
Existing	0.3234	0.1099	*0.2080*
High-Precision	*0.5487*	*0.1319*	0.2020

Table 2.2: Results reported by Bilotti and Nyberg [7, Table 1]

In [10], the authors find that the improvement in sentence retrieval using semantic roles is more salient when the predicate-argument structure of the sentence is complex; therefore, structures that contain many predicate and many arguments are more likely to take advantage of the semantic annotation.

The study in [7] does not find a correlation between retrieval performance and QA results. The authors report that a significant improvement for the retrieval set in MAP does not lead to significant improvements in QA (in terms of accuracy[20] and MRR). Table 2.2 shows the results that have been reported by the authors. Bilotti and Nyberg conclude that QA systems need greater coordination between the IR and answer extraction modules for improvements to have an impact on the final QA results.

Bilotti and Nyberg [7] share our view that the IR components of QA have historically been treated as a black box in the overall QA framework, and that these components require larger integration into QA to be able to contribute towards the overall QA performance. In a pipeline setup, as commonly employed by QA systems, the errors of each module will be carried through and contribute to the failures in the system; the authors contest the assumption that improvements in such modules will lead to improvements in QA. However, we believe as reported in Pizzato et al. [112] that the standard metrics of evaluation of IR systems do not take into account all factors that influence QA results. The presence of an answer string and supporting evidence for an answer is not enough to characterise its extraction by an QA system. Therefore, we are aligned with the view that IR and

[20]The proportion of questions which the first answer returned by the QA system is correct.

answer extraction modules must have better ties for the improvements in IR to be felt in the extraction module. Bilotti and Nyberg specify this stronger connection as bidirectional by stating that the IR must understand what the answer extraction component can and cannot extract as an answer; and the answer extraction module must keep up with the IR technology. However, the latter assumes that answer extraction is still a underdeveloped research area, which is not necessarily the case, particularly in advanced QA systems.

Bilotti et al. [10] claim that the quality of annotation influences the quality of the retrieval. The experiments show slightly worse results for the retrieval of sentences using imperfect annotations than when using gold standard annotations. However, we believe that there is a problem with the methodology because the results are not directly comparable, thus rendering their claim unsupported. The evaluation regarding the impact of the quality of the annotation in the retrieval performance use the same corpus in its gold standard annotation (PropBank [107]) and in its ASSERT [115] annotation, which is reported as 88.8% accurate. Because of the lack of questions and judgement set of sentences for the corpus used, the authors have automatically created a set of questions and judgements using the argument structure of the semantic roles. Different questions are generated for every sentence in the corpus: one question for each argument that will be omitted in the question and can be considered as an answer to it, plus one question asking for the confirmation of the whole information. The example of Figure 2.6, extracted from [10], illustrates these types of questions.

The authors have discarded all questions that appear only once, and using the sentences that created those questions as supporting sentences as the *Original Judgements*. They have also created a subset called *Reduced Judgements*, which contains the *Original Judgements* stripped of sentences that ASSERT could not correctly annotate. The *Reduced Judgements* are used to evaluate the relevance of sentences retrieved using the imperfect corpus of ASSERT annotation. Its construction is justified because the QA that will use such sentences will not be able to extract the answer from sentences

Sentence:	[$_{arg0}$ Dow Jones] [$_{pred}$ publishes] [$_{arg1}$ The Wall Street Journal, Barron's magazine, other periodicals and community newspapers]
Questions:	• [$_{arg1}$ What] does [$_{arg0}$ Dow Jones] [$_{pred}$ publish]? • [$_{arg0}$ Who] [$_{pred}$ publishes] [$_{arg1}$ The Wall Street Journal, Barron's magazine, other periodicals and community newspapers]? • Does [$_{arg0}$ Dow Jones] [$_{pred}$ publish] [$_{arg1}$ The Wall Street Journal, Barron's magazine, other periodicals and community newspapers]?

Figure 2.6: Questions automatically generated from a semantic role structure [7]

that do not have correct annotations. Therefore, the authors conclude that those sentences are not relevant. However, the definition of relevance must be global across different experiments for the results to be directly comparable. The reported difference between the gold standard and the degraded retrieval is not only minute, but may also be due to the usage of the reduced set of relevance judgements for the degraded retrieval experiment.

In this thesis, we propose a semantic annotation scheme, which is simpler and less precise than the current annotation tools that are based on PropBank-style semantic roles. We demonstrate the usefulness of such structures in IR and QA, even though the quality of the annotation is far from optimal.

2.4.2 Semantic Role Labelling

According to [87], SRL deals with sentence-level semantics, which is concerned with determining *who* did *what* to *whom*, *where*, *when* and *how*. The SRL task automatically defines the semantic relation among a predicate and their associated participant and properties, where the relations are predefined by a list of semantic roles, such as *Agent*, *Patient*, *Location*, and

[$_{\text{Cook}}$ Matilde] **fried** [$_{\text{Food}}$ the catfish] [$_{\text{Heating\_instrument}}$ in a heavy iron skillet].

Figure 2.7: Example of a sentence in FrameNet [123, page 5]

[$_{\text{arg0}}$ Matilde] [$_{\text{rel}}$ **fried**] [$_{\text{arg1}}$ the catfish] [$_{\text{arg2}}$ in a heavy iron skillet].

Figure 2.8: Example of a sentence as represented by PropBank

Manner.

The construction of SRL was facilitated in recent years because of the availability of corpora annotated with this type of information. SRL represents the relations between a predicate and their arguments. The way SRL represents these relations varies according to which standard is used. Some studies, such as the Berkeley FrameNet Project [5] are based on the frame semantic theory by Fillmore [39]. In this study, the roles of each predicate are defined based on the type of semantic frame. For instance, Figure 2.7 shows an example sentence for the semantic frame *Apply_heat*, which contains some specific semantic roles, such as *Cook*, *Food*, and *Heating_instrument*.

Gildea and Jurafsky [48] describe the first SRL tool, which is based on the FrameNet corpus. Their system performs well using the known frames of FrameNet; however, precision significantly worsens when dealing with unseen semantic frames. Because a complete set of semantic frames is difficult or even perhaps impossible to obtain, a corpus such as FrameNet is difficult to scale up for larger needs, and it has been of limited use for more general tasks.

Semantic frames are not used in PropBank [107], a corpus which uses relations between predicate and arguments without properly specifying the nature of the relation. Semantic roles are generalised by a numbering convention where the numbers should be the same for similar roles of the same predicate. For instance, argument label 0 (*arg0*) represents the agent of an action, and argument label 1 (*arg1*) represents the patient or the theme of the predicate. Figure 2.8 is an example of a possible PropBank representation of the sentence of Figure 2.7.

PropBank was manually annotated according to the PropBank Marking Guidelines [3]. The guidelines represent an effort to build a consistent set of relations; however, a closer look at the corpus shows that consistency is a hard task to achieve, particularly with the vaguely defined arguments number 3 onwards. For those cases, the inclusion of a function tag[21] proved to be useful.

PropBank provides the basis for many SRL tools [71, 142]. Even though these tools have proven to be quite accurate, they still demand a computational power that is not yet available to most research groups when working with large corpora. In our experience when using a SRL tool such as SwiRL [142], the annotation of a 3 GB corpus such as the AQUAINT [50] can take a few years of computational power[22].

In order to efficiently process a corpus with semantic relations, we have developed an alternative annotation strategy based on word-to-word relations instead of noun phrase-to-predicate relations. A similar approach using semantic relation triples is taken by Litkowski [77, 78, 79, 80] in his QA system. The next section focuses on this work from Litkowski.

2.4.2.1 Semantic Triples at CL Research

Litkowski's participation in the TREC QA track from 1999 to 2002 (TREC-8 to TREC 2002) involved the creation of semantic triples from sentences and questions, and the alignment between sentence triples and question triples to find the answer for the questions. A database is populated by the semantic triples generated from the top-n documents[23] provided by NIST. The QA process involves creating semantic triples for questions and searching the database for the same triples and for an element that matches an unbound variable in the question's semantic triple.

[21] A function tag is information attached to the arguments that represents relations such as negation, location, time and direction.

[22] Intel(R) Pentium(R) 4 HT 2.80GHz with 2.0 GB RAM

[23] Litkowski uses the top-10 or top-20 documents depending on the year and the task. The number of documents seems to be arbitrary.

[Matilde $_{\text{SUBJ}}$ fried] [the catfish $_{\text{OBJ}}$ fried] [a heavy iron skillet $_{\text{IN}}$ fried].

Figure 2.9: Sentence of Figure 2.7 as semantic relation triples of Litkowski [77]

The semantic relation triples use surrogate terms to represent the semantic roles such as *SUBJ*, *OBJ*, *TIME*, *NUM*, *ADJMOD* and the prepositions heading prepositional phrases. Figure 2.9 demonstrates the representation of the sentence of Figure 2.7 as Litkowski's semantic relation triples. Triples involving appositive phrases are added in Litkowski's TREC-9 participation [78].

In [78] improvements are added to answer specific types of questions such as *Where* questions. The matching rules for the question types are manually adjusted with the knowledge previously acquired in TREC-8 [77]. These handcrafted heuristics, which are based on syntactic clues, have been proven to be able to create semantic triples to match a question to its answer. We believe that these handcrafted heuristic can be augmented by automatically acquired rules. This thesis studies whether an automatic method to generate semantic triples can be used to populate an entire IR database for a large set of documents. This database of semantic triples should be able to provide documents that are more likely to contain answers to questions than the standard linguistically uninformed method.

2.4.3 Structured Retrieval

As discussed previously, researchers have achieved varying level of success with applying linguistic information to search engines [7, 140, 146]. It is hard to establish a direct comparison between the approaches because experimental setups differ considerably in terms of corpora, NLP tools, linguistic information, indexing and retrieval strategies.

An intuitive approach towards encoding linguistic information is through modifications in the dictionary structure [2, 140, 146]. The dictionary modification approach is appealing because it is easy to implement with existing

off-the-shelf tools. This is also applicable to XML retrieval systems that modify existing IR tools, such as the JuruXML system by Mass et al. [88]. We have also experimented with such an approach, however we noticed that it does not scale well, leading to poor overall speed performance. The main reason for the poor performance is the increase of the vocabulary size, which forces the data structure, which is normally maintained in main memory, to use disk storage. As the size of the vocabulary increases, the average of posting items per vocabulary item decreases. This causes a large number of terms in the dictionary to point to single items in the posting list, which leads to a posting list that is inefficient and difficult to compress.

Bilotti et al. [8, 10] specify a retrieval methodology for semantic roles using XML with the Indri Search Engine [138]. The literature on the function and architecture of Indri does not clearly address the way that the XML information is encoded; an early publication on Indri's field-indexing feature specifies that different inverted files are built for each field type [90]. It is also noted that the method is extended to include hierarchical overlapping fields in [105], but it does not make a clear description of the data structures used.

Our work, which will be described in Chapter 4, adopts a related approach to the work of Fuhr and Gövert [43, 44] and to the work of Carmel et al. [21], which we will describe next.

2.4.3.1 Index Representation at the University of Dortmund

Fuhr and Gövert from the University of Dortmund describe an inverted file approach that encodes structured information from XML files with minimal redundancy [44]. They call this method the *PIL* approach (path in inverted lists). In *PIL*, every posting item contains all the information regarding its XML context, which includes the complete XML path and the indices to the XML structure. The posting list of the *PIL* approach is made of the following items:

- *posting list*, which contains a list of *document entries*;

```
< 34 3
  < 4 <book chapter section #PCDATA> <1 1 1 1> <1 3 1 1> >
  < 6 <book chapter section list item #PCDATA> <1 1 1 1 1 1> <1 3 1 2 1 1> >
  < 6 <book chapter section list item #PCDATA> <1 1 1 1 2 1> <1 3 1 2 2 1> >
>
< 40 2
  < 4 <book chapter section #PCDATA> <1 2 1 1> <1 4 1 1> >
  < 4 <book chapter section #PCDATA> <1 2 2 1> <1 4 2 1> >
>
```

Figure 2.10: The posting list of the *PIL* approach (excerpt from [44, page 4])

- *document entry*, which contains a *document id*, a *number of occurrences*, and a list of *occurrences*; and
- *occurrence*, which contains a *path length*, a list of *XML elements*, a list of *elements indices*, list of *elements sequences*, and an optional *weight*.

The authors demonstrate the usage of the *PIL* approach using a book structure and the example of a term that occur in two documents (34 and 40). The example shown in Figure 2.10 (excerpt from [44, page 4]) uses brackets and spacing to help identify the items of the previous list.

A straightforward implementation of *PIL* would require vast amounts of disk space; hence, some strategies are applied to compress the inverted list. Fuhr and Gövert's approach focuses on reducing redundant structures, such as repetitions in the lists contained in *occurrences*. The specific details of these lists are not important for our understanding, what is important to note is that *path length*, which controls the amount of items in these lists, can be modified to encode only the changes in the structure. This leads to much smaller structures, since all lists will be reduced in size and the posting list will not encode redundant information.

Our indexing method to be presented in Chapter 4 adopts a similar approach to the compressed version of *PIL*, which only encodes the changes in the structure for the different post of the same item. Unlike XML retrieval, our method is optimised to work with fixed structures specified by linguis-

tic relations, therefore we can achieve higher inverted file compression, and better indexing and retrieval speeds.

2.4.3.2 Vector Space Extension at IBM Research Lab and Haifa University, Israel

Carmel et al. from the IBM Research Lab and Haifa University in Israel propose a retrieval model for XML that allows partial matches and ranked retrieval with the use of XML fragments [21, 22]. They propose an extension to the vector space model where the dimensions of the document and query vectors are not restricted to terms only. The dimensions are made of pairs in the form of (t_i, c_i) where t_i is a term and c_i is the context of this term. In our work, to be described in Chapter 4, we also propose an extension to the vector space model that works with different dimensions beyond terms.

Unlike data-centric XML retrieval[24], Carmel et al. allow inexact matches between contexts for terms. They propose the use of a vector space model that expresses context resemblance, which defines whether a certain context c_j is similar to a context c_k. Context resemblance $cr(c_j, c_k)$ is a measure between 0 and 1, which is only defined as 1 when $c_j = c_k$.

The value for the context resemblance is calculated from a combination of criteria including the Longest Common Subsequence (LCS), average position of the optimal match between the two contexts, LCS with minimum gaps between contexts, and length difference. All calculations are normalised and given arbitrary weights. More importantly, the value computed for context resemblance is used to calculate the relevance $\rho(D, Q)$ of a document D to a query Q in an augmented vector space model as shown in Equation 2.12:

$$\rho(Q, A) = \frac{\sum_{(t_i,c_i)\in Q} \sum_{(t_i,c_k)\in D} WQ(t_i, c_i) * WD(t_i, c_k) * cr(c_i, c_k)}{|Q| * |D|} \quad (2.12)$$

[24]XML is defined as data-centric when the structure and attribute data is predominant over textual data. Search in this context is similar to how queries are executed in relational databases. RDF triple retrieval [167] is an example of a predominant data-centric retrieval task.

Where $WQ(t_i, c_i)$ is the weight of the pair (t_i, c_i) in query Q, and $WD(t_i, c_k)$ is the similar measure for document D.

2.5 Concluding Remarks

In this chapter, we have presented the required background information to enable a sound comprehension of the rest of this thesis. We introduced an initial concept of IR with the vector space model and the evaluation metrics. We also discussed QA and how it has evolved in recent years. We have presented the QA framework and how researchers have addressed some specific problems. One of these problems is IR for QA, which tries to find an optimal way to reduce the number of textual instances that contain the answer to the question.

The last part of this chapter involved the analysis of some studies that are directly related to our research for the development of this thesis. Our thesis is a reflection and a continuation of the work of these researchers. The work by Tiedemann [146] includes several types of language information in the IR of a QA system; they have shown improvements in IR using QA metrics, which does not necessarily mean that the improvements will flow on to the final QA results. For example, Bilotti and Nyberg [7] apply semantic role information in the IR stages of a QA system to create a high precision retrieval set, however these improvements did not produce a significantly positive impact to the overall QA results.

Despite the discouraging results of Bilotti and Nyberg [7], it is too early to conclude that better QA cannot be obtained by enhancing IR. We believe that improvements can be achieved in the QA pipeline with improvements in IR. Our experiments are based on a semantic annotation that is simpler, and because it is quicker to create than SRL, it is better suited to the IR requirements. Our semantic annotation follows on from the work of Litkowski [77]. Litkowski proposes a semantic model based on dependency relations that is used as the main component for extraction answers in his QA system. In our work, we propose an IR model that is able to represent a wide range

of linguistic relations. These linguistic relations are used similarly to how context is used by Carmel et al. [21, 22] to create a modified vector space ranking that considers dimensions not limited to terms. Our indexing strategy is similar to that proposed by Fuhr and Gövert [43], who describe an indexing method for structured documents.

3

Pseudo-Relevance Feedback Based on Expected Answer Type

Probabilistic IR, as described in Section 2.1.1, estimates the probability of relevance of documents with respect to an information need expressed by a user-provided set of keywords. The estimation of these probabilities is often assisted by the information contained within documents that are known to be relevant for each specific query. The technique of informing the IR system which documents or information are relevant to a specific query is known as relevance feedback. Relevance feedback techniques have been used for many years, and they have been shown to improve most probabilistic models of IR [124].

Relevance feedback is considered to be pseudo (or blind) relevance feedback when there is an assumption that the top documents retrieved have a higher precision and that their terms represent the topic expected to be retrieved. In other words, it is assumed that the documents at the top of the retrieval list are relevant to the query, and information from these documents is extracted to generate a new retrieval set.

Factoid questions such as "*Who is the president of Brazil?*" and "*What is the capital of Brazil?*" require named entities as their answers. In this chapter, we explore this characteristic of factoid questions and study whether the presence of named entities are a good indication of relevance in retrieved

documents. We describe an experiment that uses a pseudo-relevance feedback technique that uses named entities as an indication of relevance applied over a probabilistic IR system to try to improve QA performance. We assume that documents that address the correct topic but do not contain any named entities of the expected answer class would have a low probability of relevance regarding QA. We hypothesise that documents that contain named entities of the correct class have higher probability of relevance than documents that do not.

The relevance feedback applied to QA differs from the one applied to general IR in that QA deals more with the presence of a passage that can answer a certain question, rather than with the presence of its topic. Hence, our technique focuses on feeding terms into the IR engine that could represent an answer for the questions. It is possible to apply the technique to all question types that can be identified by the named-entity recognizer, however in this study we pay particular attention to the results of questions regarding personal names. We acknowledge that other question types are just as important and may generate different results due to the different frequencies of their appearance in the documents; however personal names can provide us with concrete results since it is a type of named entity that has been widely experimented on recognisers and is likely to be present in most types of newswire texts, such as those in the AQUAINT corpus [50]. We performed our experiments using the AQUAINT corpus and the question set from the QA track of TREC 2004 [161].

It must be noted that the results we show in this study represent only one class of questions, and that the pseudo-relevance feedback mechanism is highly dependant on the question classification module. Section 3.1 provides some background information on the question classifier that we use with the relevance feedback technique. Section 3.2 explains the principles behind the named-entity relevance feedback technique and how we implemented it. Section 3.3 focuses on the evaluation of the technique regarding its use as an IR tool and as a module of a QA system. Section 3.4 presents the concluding

remarks and future work.

3.1 Trie Classifier

We developed a question classification system based on trie structures. A trie $T(S)$ is a well-known data structure that can be used to store a set of sentences in an efficient way. It is defined by a recursive rule

$$T(S) = \{T(S/a_1), T(S/a_2), \ldots, T(S/a_r)\}$$

where S is a set of sequences (sentences, in our case). S/a_n is the set of sequences that contains all sequences of S that start with a_n, but stripped of that initial element [27].

In each node, local information extracted during training is stored. This includes the word, class and frequency information. Since each node represents part of a unique path in the trie, frequency information is the number of sentences that use that particular node in a path in the trie.

Figure 3.1 illustrates the trie containing the following questions:

- "*Who is the President of Brazil?*",
- "*Who is the Brazilian football coach?*",
- "*What is the name of the Brazilian President?*",
- "*What is the capital of Brazil?*", and
- "*Where is Brasilia located?*".

Extra information is stored per node, so for instance, the node that is reached by following the path *What* contains personal name and location as answer classes, both with frequency of one.

The classification of new sentences is performed by extracting the class information stored in the trie nodes. The words of the new sentence are used to find a path in the trie. The class information can then be extracted from the final node of the path through the trie. Our novel question classifier

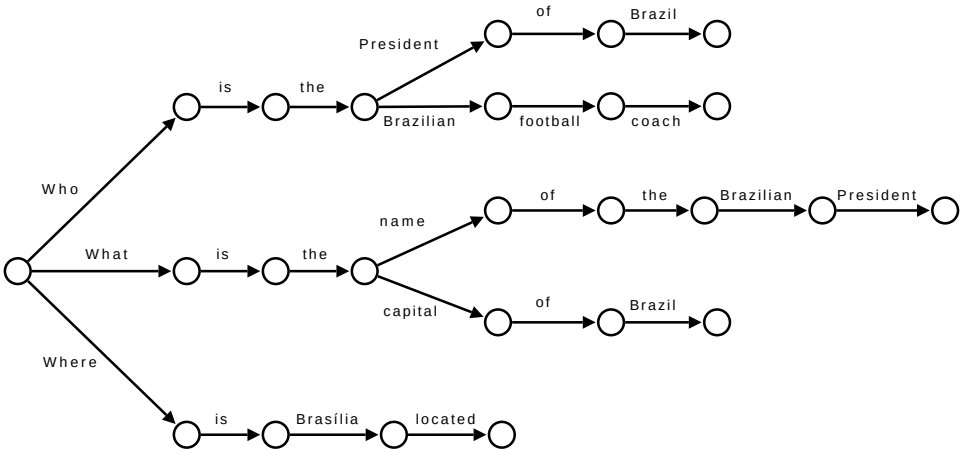


Figure 3.1: Example of a trie representation of questions

method is described in some of our published work [110, 171, 172]. In a comparison to the results of Zhang and Lee [174], who use the same question data to train various machine learning methods, we have obtained slightly worse results when applying a much simpler methodology.

3.2 Relevance Feedback Using Named Entities

Because most fact-based questions require answers to be named entities, we hypothesise that a relevance feedback mechanism that focuses on this type of information can help IR for QA. We adopt the QA concept of relevance by trying to eliminate those documents that would not be able to answer a factoid question. Thus, the process will guide the document retrieval towards documents that are relevant to the question topic (general IR relevance) and particularly towards those containing entities that could answer the question (QA relevance).

Suppose for question Q on topic T[1], we can use a probabilistic IR engine PIR over a corpus C to obtain a retrieval set of documents R_1:

$$R_1 = PIR(Q + T, C)$$

[1]In TREC QA Tracks [161] questions are centred around a certain topic, for instance the topic *Australia* could have a question such as "*Who is the Prime-Minister?*"

where $Q + T$ is a IR query composed of Q and T combined. Our process applies a named-entity recogniser NER over the top-n ranked documents of R_1, thus obtaining a set of named entities E:

$$E = NER(R_1, n)$$

The feedback process consists of enriching the previous query as $Q+T+E$ in order to obtain a new set of documents R_2:

$$R_2 = PIR(Q + T + E, C)$$

The query $Q+T+E$, which was used to retrieved R_2, is the concatenation of Q, T and E with or without different weightings for the different query components.

Our expectation of this technique is that not only documents that contain the correct answer in R_1 will be boosted in ranking on R_2, but also that documents that have a high ranking in R_1 and do not contain any named entity of the EAT will be demoted in R_2. Therefore, documents that would not theoretically contribute to the QA performance will not take part on the answer extraction phase, allowing their slots of processing time to be occupied by other more relevant documents.

As an example of this process, consider the TREC 2005 QA Track Question 95.3 on the topic of *the return of Hong Kong to Chinese sovereignty*: "*Who was the Chinese President at the time of the return?*"

The first phase of the process is the question analysis, which defines what is the EAT and what are the question main words. Then the question and its topic define an IR query that generates the retrieval set R_1.

The next process extracts the named entities of the EAT from the first n documents in the R_1 set of documents. For this example, fifteen names of people were extracted, mostly Chinese names and all of them related to politics. A new IR query is built using these fifteen names and the final set R_2 of documents is retrieved.

The list of names found for this query is listed on Table 3.1. We can observe that among those names there is the correct answer for the question

Named Entities	
President Mario Alberto N. L. Soares	*President Jiang Zemin*
General Secretary Aleksandr Zharikov	Minister Qian Qichen
Minister Sabah Al- Ahmad Al-Jaber	Minister Zhou Nan
Prime Minister Mahmoud Zouebi	Mr. Deng Xiaoping
President Maumoon Abdul Gayoom	Premier Li Peng
President Ugo Mifsud Bonnici	Liu Huaqiu
President Meets Chinese	laws Will
President Leonid Kuchma	

Table 3.1: Extracted named entities

(*President Jiang Zemin*), which helped to generate a better retrieval for this question with the pseudo-relevance feedback mechanism.

Our hypothesis is that the named-entity feedback technique improves the overall document retrieval for QA by providing a retrieval set of documents that facilitates the extraction of the correct answer by a QA system. The technique should theoretically improve good questions (where a correct feedback is obtained) and not deteriorate bad ones. A question is 'bad' if we cannot retrieve any document that contains an answer in R_1.

3.2.1 Implementation

The technique consists of posting the original question to a probabilistic IR engine, extracting the named entities of the EAT from the top-n results, and re-feeding the IR engine with an expanded query. By doing this, we are informing the IR system that documents containing those named entities are relevant to the question. Several implementations and setups can be tested using this approach; we implemented a basic framework as shown in Figure 3.2.

We developed our IR system using C++ and the XAPIAN Toolkit for Probabilistic IR. The AQUAINT Corpus [50] was indexed using full text

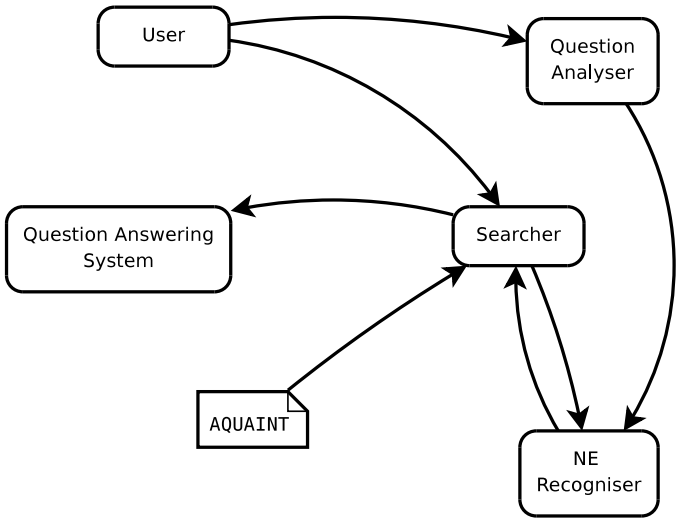


Figure 3.2: System overview for the relevance feedback process

without stopwords, and it was searched using Xapian Probabilistic Methods (which uses Okapi BM25 [120] for ranking).

As depicted in Figure 3.2, the user poses a question to the system, which is simultaneously processed by the question analyser and the searcher. The question analyser returns the EAT (a named-entity class for factoid questions), while the searcher returns a list of documents or snippets of text from the AQUAINT corpus ranked by the Xapian BM25 implementation. The named-entity recogniser receives the output of these two processes and extracts the corresponding named entities from the received files. Once this is done, it re-feeds the query to the searcher with the additional named entities. The searcher then feeds the results into the QA system.

3.3 Experiments and Evaluation

In our experiments we used the data collection made available by NIST for the TREC QA Tracks[2]. All the questions and judgement files of the TREC 2003 QA Track were used on a preliminary evaluation of this process. Because this experiment required that all components shown in Figure 3.2

[2] `http://trec.nist.gov/data/qa.html`

to be fully functional, several setups were implemented, including a simple passage retrieval algorithm.

In our evaluation, we labelled the documents as relevant or not relevant by assuming that relevant documents are those containing the required answer string. These early tests showed us that using the set of 500 TREC 2003 questions with our pseudo-relevance feedback technique improved the results over the initial retrieval. The improvement, however, was small and not statistically significant.

In our system architecture, the question classification was performed using the Trie-based technique [171] which has a performance of around 85% as previously described in this chapter. This means that in 15% of the cases, we might have an immediate degradation of the results (by adding the wrong named entities to the query). To ensure complete correctness on this part of the module, we trained the classifier with the same questions as the verification set. However, because of the large amount of expected answer types present in the classification we used, named-entity recognition proved to be a particularly complex task. For this purpose we used two well known named-entity recognisers: Lingpipe[3] and ANNIE[4] from Gate.

We also observed that some named-entity classes could not be properly identified by our named-entity recogniser. Therefore, we shifted our attention to only people's names, as we understood them to be less likely to suffer from this issue. Because of this, we have manually selected all factoid questions that required some person's name as the answer for the evaluation of our system.

The evaluation was performed intrinsically and extrinsically as Spärck Jones and Galliers [63] describe. Intrinsic and extrinsic evaluations differ because the former evaluates a system according to its primary function, while the latter evaluates a system according to its function or its setup purpose. In our study, the evaluation was performed using the combined set of questions and topics of the TREC 2004 and 2005 along with their respective

[3] `http://www.alias-i.com/lingpipe`

[4] `http://gate.ac.uk/ie/annie.html`

judgement sets. We experimented with different setups, but the variations were primarily with the passage window, the number of top documents used and the weights assigned to the different components of the query. However, the results were not affected by the variation of weights, therefore we only present the results where the different query components are weighted equally. We extrinsically evaluated the effectiveness of the retrieval sets by the percentage of correct answers that the AnswerFinder (AF) [96] system generated, and intrinsically evaluated the same sets of documents using precision, coverage and redundancy.

We applied the retrieved document set to AF and measured the exact results using the patterns made available by Litkowski on the TREC QA Data Web page.

3.3.1 Results

Our evaluation focused on using pseudo-relevance feedback to enrich the IR query used by QA systems to find some documents that could answer natural language questions. We retrieved sets of documents using a combination of topics (T), questions (Q), entities (E) and answers (A). The following combinations were tested:

T: Only the topic T is sent as a query. This set of queries evaluates the potentiality of improving the retrieval set that NIST provides for every topic.

TQ: The queries are made of the topic T and question Q. This is the current retrieval set used by AF.

TQE: This is the feedback technique, where topic T, question Q and the named entities E extracted from top-n documents are combined, which might or might not contain a correct answer.

TQA: This is the optimal feedback technique, where topic T, question Q and answers A are combined. This set evaluated how far we are from

the optimal retrieval. The answers A are obtained from the evaluation scripts of TREC's QA tracks.

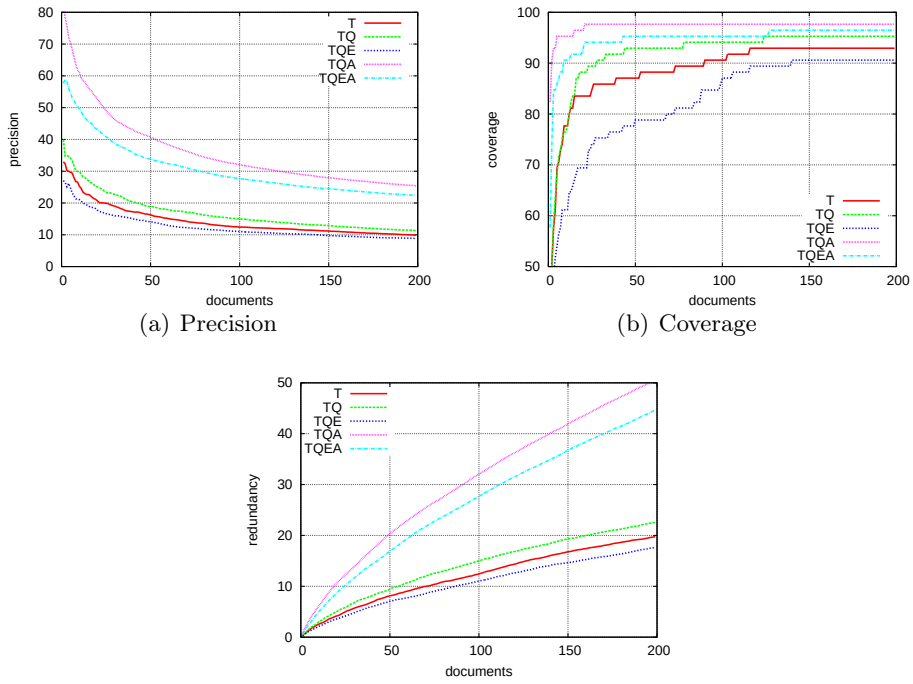
TQEA: These queries combine the feedback technique with the answers, so we can measure the amount of noise introduced by adding named entities that are different to the answers. We made sure that named entities that were equal to an answer were not introduced twice so their score would not be erroneously duplicated.

Different combinations could also be tested, for instance TA, TE or just A, E and Q. We understand that those and other combinations could provide some insight on certain matters, but we believe that they would not represent a realistic retrieval set. It is a fact that the terms from T must be present in the retrieval set, since all documents must address the correct topic. For instance, including Q without having T will not generate a relevant retrieval because the subject of the question is not present. Also, including A or E without Q and T may represent a completely different retrieval that is not desired in this study.

The precision, coverage and redundancy obtained for the TREC 2004 and 2005 questions regarding people's name are respectively shown in Figures 3.3(a), 3.3(b) and 3.3(c). We note that the results for the feedback technique do not improve the results on neither T nor TQ on any of the measures we obtained. As expected, the addition of the answer (TQA) represents the optimal retrieval set, obtaining 86% coverage on the first document per question and over 90% coverage on the second.

The noise introduced on TQEA is not a major concern when the answers are involved in the query. This is an indication that most entities found by the feedback mechanism do not represent an answer. This raises two questions: how to improve the technique so that the answers are included in the feedback; and how to minimise the noise so that potentially good feedback is not worsened.

To address the first problem we can foresee two solutions: one is improving the accuracy of the named-entity recogniser; something we do not address



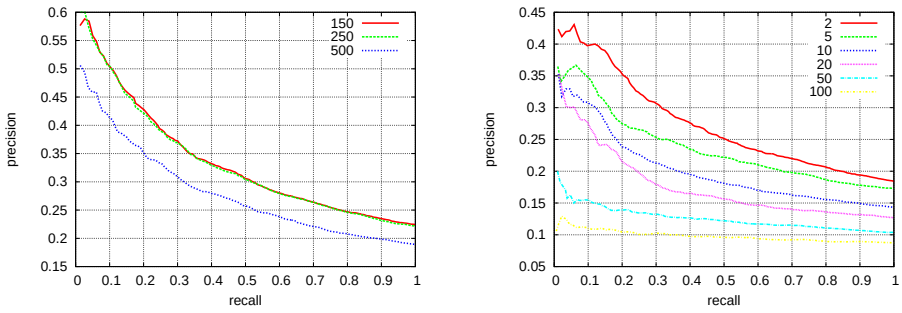
(a) Precision (b) Coverage

(c) Redundancy

Figure 3.3: Precision, coverage and redundancy for questions about people's names of TREC 2004 and TREC 2005

in this study. The other is to increase the search space without adding more noise in the query. This is a difficult task and it could be achieved by finding the smallest possible windows of text containing the answer. We performed some experiments using different numbers of documents and variable passage size: for now, we have found that fewer documents and smaller passages provide the best results. Figure 3.4(a) demonstrates these experiments using three different window sizes and the results of the $TQEA$ queries. It can be observed that the results were similar for windows of 150 and 250 characters, with a substantial drop in the average precision when using windows of 500 characters. Figure 3.4(b) shows that results worsen when increasing the number of documents used to extract the named entities.

We understand that documents in the first retrieval set R_1 will contain named entities of the same type, but not necessarily the correct one (the answer), thus creating some noise in the query. We believed that a certain



(a) Different window sizes

(b) Different numbers of documents to extract named entities

Figure 3.4: Precision and recall for $TQEA$ queries

degree of noise would not hurt the retrieval performance. However, our experiments, as shown, demonstrate otherwise. The noise created by erroneous entities affects the performance once the elements in E become more important than the elements in Q. Because we cannot guarantee the correctness of any of the named entities included in E, the resulting retrieval set R_2 might represent a worse retrieval set than R_1. However, these cases may not influence the results in a QA system since R_1 would also not lead to the correct result.

This shows that our feedback technique suffers from the same flaws most pseudo-feedback techniques have. For instance Ruthven and Lalmas [124] show that when the initial retrieval set is not good, the pseudo-feedback techniques is likely to worsen the results because, instead of bringing the query closer to the topic at hand, it will take it further away (a phenomenon called query drift). We hypothesise that since our technique is meant to be applicable over a QA system, if the initial set of results is bad (i.e. it does not contain the answer), there is not much that can be worsened. To confirm this hypothesis, it is necessary to perform an evaluation over a QA system. Table 3.2 shows the runs of QA performed using the same set of questions of the intrinsic evaluation and the documents retrieved by the retrieval sets shown before.

Run	Exact
T	19.6%
TQ	28.6%
TQE	23.2%
TQA	28.6%
TQEA	*32.1%*

Table 3.2: Correct answers on AF

What can be observed here is that the feedback technique (TQE) offers a better set of documents than the technique which only uses topics (T). However, they are still worse than the topic and question technique (TQ). An interesting result is that $TQEA$ gives the best results, which suggests that the addition of entities helped to retrieve sets of documents that are easier to process by the QA system. Even though our process did not show improvements over the baseline techniques, it was very important to find that the results of the intrinsic evaluations of the IR component do not parallel the results of the extrinsic evaluation of the QA system. Despite the fact that high precision, coverage and redundancy represent a better chance of finding answers, we showed that they do not guarantee a better performance over a QA system.

Comparing the results of T and TQ we observe that they are very similar on the intrinsic evaluation and quite different on the QA system. Therefore, what appears to help QA is the presence of more context words so that the answers not only appear in the document, but are also present in the context of the questions. This is mostly due to the fact that QA systems tend to work with full discourse units, such as sentences and paragraphs, and the selection of those are normally based on words from the topic and the question.

In summary, our experiments did not confirm the hypothesis that named-entity feedback would help improve QA. But, in the ideal situation where the answers are identified and included in the queries, the improvements are

clear under an intrinsic evaluation. The differences between the intrinsic evaluation and extrinsic evaluation highlight that there are many issues that IR evaluation metrics do not currently cover.

3.4 Concluding Remarks

In this chapter, we have examined whether a pseudo-relevance feedback mechanism could help the QA process, with the assumption that a good indication of a document's relevance for its usage in aQA system is the presence of named entities of the same class required as the answer for a certain question. Our assumption was based on the fact that documents not containing those entities are less likely to help provide the correct answer and every entity of the right type has a probability of being the answer.

Unlike the standard method of discarding documents that do not contain entities of the EAT, which is normally done at the answer extraction phase of the QA systems, our method uses the named entities of the EAT as part of a query. This modifies the document list not only in terms of the named-entity distribution, but also in the order that the documents appear. Our expectation was that documents that contain the answer but do not appear at the top of the first retrieval set would be boosted in rank at the second retrieval. We have shown that we can obtain a better retrieval set if the answer is present in the modified query; however the lack of the answer in the top results heavily penalises the overall retrieval set by drifting the topic of the query away from the question topic.[5]

We have described our evaluation of the hypothesis using known IR metrics and a QA system. Our main conclusions are as follows:

- Because we have not yet reported satisfactory results, we believe that although the method is intuitive, it will not produce good results unless

[5]This indicates that we could obtain a better pseudo-relevance feedback technique with an improved IR, however we believe that Xapian provided us with the state-of-the-art in keyword retrieval and little benefit would be obtained by experimenting with different IR tools.

more sophisticated control over the introduced noise is achieved; and

- The evaluation of the technique brought to our attention the fact that it is not possible to state that a retrieval technique is better just by relying on conventional IR evaluation metrics. The differences on the intrinsic and extrinsic evaluations demonstrate that there are many hidden variables that are not taken into account in metrics such as precision, coverage and redundancy.

Therefore, the investigation of our first research question could not show that named entities provide strong evidence of relevance for documents in IR for QA.

4

Linguistically Motivated Indices

Question Answering (QA) is the task of finding information in a large repository of documents. The process of information search is done via textual queries, which represent the information needs of a user, and which can be presented textually as a full natural language description or question, or simply as a set of keywords. Even for a small document collection, it is impractical to search for these keywords by reading all of the documents. In order to speed up the information search IR techniques build a data structure called an inverted file.

The classic representation of an inverted file is a list of words where every word has an associated list of documents [86, page 6]. This representation, which we introduced in Chapter 2, allows a quick search by providing fast access to the list of documents for every query word that can be found in the documents. With a list of documents and the statistics of term occurrence at hand it is possible to use different ranking strategies to find the documents that are more important according to the query words used. These ranking strategies have the strength of being language independent; however, most of them simply consider documents as a bag of disconnected words. It is a fact that words do not appear in a natural language document by chance; on the contrary, words are sequenced and formatted in order to form a meaning and express concepts. In this sense, it is expected that a strategy that encapsulates the linguistic relations among the document terms will provide

the means for more sophisticated retrieval strategies.

As described in the literature review in Chapter 2, NLP can assist IR when the Bag-of-Words (BoW) approach is not enough to distinguish the user's need. Specifically, when the BoW results are too restrictive or too general, NLP can be used to find the right set of documents. Such cases can involve the use of word-sense disambiguation, syntactic dependencies, semantic roles, lexical substitution or other NLP techniques in both queries and documents.

Even though NLP can have a clear impact on IR results, the IR community has not yet seen a clear cost-benefit for adding support for such techniques. The lack of support for NLP information from IR tools makes it hard for researchers to integrate these techniques. In this chapter, we propose a multi-layer inverted file representation that allows the inclusion of different linguistic information, such as part-of-speech, syntactic relations and semantic roles. These techniques form the basis of the Tupi framework and the Jemu IR system, which are described in Appendix A. First, we describe the multi-layer inverted file structure, and then we present the possible representations and some experiments. Finally, we present some performance evaluations in regards to disk space allocation, and the speed of indexing and retrieval.

4.1 Representing Linguistic Information in Information Retrieval

IR systems base their search algorithms on the existence of words from queries expressed by a set of keywords. A keyword search means that the token granularity of the search is at the word level. Different retrieval tasks may require different token granularity. For instance, a word search tool based on specific sound descriptions could use IR techniques similar to keyword search but using phonemes instead. The choice of the type of token to use is defined by the type of task and corpus. For instance, the usage of

words as the unit for indexing in cross language IR can be hazardous when dealing with morphologically complex languages. For such tasks, the usage of morphemes is preferable as shown by Chew et al. [25]. They describe a novel approach to latent semantic analysis [34] that utilises morpheme-like units calculated using mutual information of character n-grams.

Most search queries can be satisfied using current indexing and search techniques, which explain the good results for such tasks. However, a few queries are still hard to solve when considering the text and queries as BoW. Take the example of the query: "*comments about the election candidates for the US presidency*". A BoW approach will likely to return a list of relevant documents as well as a list of irrelevant documents about:

(1) the election candidates for an unrelated presidency (other countries or US organisations);

(2) the US presidential election process; and/or

(3) comments made by the candidates, not about them.

Because of the high volume of documents dealing with the US presidential election and candidates, it is expected that documents in item (1) will not appear at the top of the ranked list of documents. However, documents in items (2) and (3) are likely to appear at the top of the list, particularly when only BoW methods are used. For these cases, a more language-oriented approach is required.

If a more precise retrieval is needed, it becomes necessary to perform a further analysis on the documents and on the queries; such analysis could include linguistic information such as syntactic dependencies and semantic roles. The addition of extra linguistic information allows the construction of a different retrieval model to the standard BoW. For instance, the inclusion of syntactic dependencies can help to distinguish between the search for "*comments about the candidates*" from the search for "*comments about the election*". In some cases, this distinction can be done by post-processing the results of the BoW retrieval; however, most retrieval sets are too large

to be processed in real time, which causes linguistically motivated post-processing techniques to be applied only to the top-n documents of the BoW retrieval set. As a consequence, a retrieval model obtained by post-processing is different to the retrieval model where the linguistic information is pre-encoded into the search structure. Thus, linguistically motivated retrieval models are only achievable if the linguistic information is added to the search indices.

Furthermore, as presented in the Introduction in Chapter 1, the addition of linguistic information into the search index can speed up systems that depend on it, particularly when linguistic information is used to further analyse the documents to find a more precise document set or portion of documents that can better satisfy certain task requirements. QA is an example of such IR dependent systems that use NLP techniques to cut down the number of candidate documents or candidate sentences that can answer a certain question.

The inclusion of linguistic information in IR has traditionally occurred through modifications over the token representation, i.e. tokens are modified from simple words to strings formed by words and the required linguistic information. Tiedemann [146] uses a special character to separate the different types of information. In his representation, a syntactic dependency relation $\mathcal{R}$ between a modifier word $\mathcal{W}$ and the head word $\mathcal{H}$, is presented as one single token $\mathcal{W}/\mathcal{R}/\mathcal{H}$ in the index. This means that a subject relation between *John* and *sings* is represented as *John*/*subject*/*sing*.[1] This type of representation is also adopted by Strzalkowski [139] for the inclusion of linguistic motivated streams such as head and modifier pairs and compound names, and similarly adopted by Carmel et al. [22] when representing context in XML retrieval. This type of representation satisfies most requirements for the addition of linguistic information, but it can lead to a large vocabulary size, and redundancy in the stored information. This problem is more salient when words share the same context in XML retrieval, or the linguistic rela-

[1]For this representation, Tiedemann [146] uses the root form of the words.

tions need to be partially represented ($\mathcal{W}/\mathcal{R}$ and $\mathcal{H}/\mathcal{R}^{-1}$), which causes the process to be slow and for large amounts of data storage to be required. The following section describes our approach taken to encode linguistic motivated indices.

4.2 Inverted Files

The standard inverted file implementation assigns document-identification numbers for every term in the document collection [86, page 6], which can be seen as a collection of tuples (t_j, d_i) where t_j is a term and d_i is a document. In order to allow a fast access for search, the file is ordered by terms, which creates a structure where every term t_j contains a list of document d_i as

$$\boxed{t_j} \rightarrow \boxed{(d_1, d_2, \ldots, d_k)}$$

This type of representation can be modified to assist the specific task that the inverted file is designed for. It can incorporate index compression techniques and distributed indexing/retrieval for a large collection of documents.

Other types of information could also be added to the inverted file, for instance if a ranking function based on Term Frequency (TF) and Inverted Document Frequency (IDF)[2] is used, it is necessary to add or to provide a quick way to obtain the frequency counts of the terms for every document. In order to speed up retrieval time we can add the frequency counts:

$$\boxed{t_j/df_j} \rightarrow \boxed{(d_1/tf_1,\ d_2/tf_2, \ldots, d_k/tf_k)}$$

In this representation, tf_i is the amount of times the term t_j appears in the document d_i, and df_j is the number of documents that contain the term t_j; for the representation above dj_j is equal to k.

As proposed by Williams et al. [168], positional information of document terms can be used as a way to retrieve phrases such as "*Government of New South Wales*" and "*George W. Bush*". With positional information, every document d_i will be associated with a set P_i^j representing all the positions of a term t_j in d_i:

$$P_w^x = \{p_1, p_2, \ldots, p_f | p_k \text{ are the positions of } t_x \text{ in } d_w\}$$

[2]See Equation (4.2) further defined in this chapter.

This strategy also helps to compute TF and IDF if they are not yet included in the index, but it has the drawback of increasing the index size.

We see the addition of positional information in the inverted file structure as a separate layer of indexing. This multi-layer representation consists of one outer layer that represents the documents where the terms appear, and an inner layer that provide the information about every position where the term occur in a specific document. In this case, there are two layers of information and the inner layer is dependent on the outer layer.

This multi-layer representation can be extended to include other types of information including nested and dependent information such as documents, sections, paragraphs, and sentences to unrelated information such as word position together with POS. Our approach to encoding linguistic information in the search index concerns the development of a multi-layer inverted file representation, which is described in the next section. This approach is implemented in the Tupi Framework and in the Jemu IR System, both described in the Appendix A.

4.3 Multi-layer Inverted File

We decided to build a multi-layer inverted file framework because of its ability to represent different types of information in a single structure. In this way, it is possible to retrieve documents using different types or layers of information using one single structure, avoiding building redundant dictionary entries or other data structures.

When layers are dependent on each other, the information added to the inverted file follows a wrapping strategy from the most general type of information to the most specific; thereby, avoiding representing redundant information. For instance, if there is a requirement to represent the documents, sections and positions of words in a document, the following would be encoded for a term t_1 that appears at positions p_1 and p_2 of section s_1 and position p_3 and p_4 of section s_2 of document d_1:

$$\boxed{t_1} \rightarrow \boxed{(d_1 \rightarrow (s_1 \rightarrow (p_1, p_2),\ s_2 \rightarrow (p_3, p_4)))}$$

This type of inverted file representation allows the inclusion of relational information between terms by treating relations as different bags of information. For instance if term t_1 has a relation of type $\mathcal{R}$ with term t_2 (the inverse relation is $\mathcal{R}^{-1}$) in document d_y we can specify this relation by representing them as

t_1	$\rightarrow$	$(d_y \rightarrow (t_2/\mathcal{R}))$
t_2	$\rightarrow$	$(d_y \rightarrow (t_1/\mathcal{R}^{-1}))$

This representation allows the retrieval of documents, relations and related terms for every indexed term with minimal redundancy.

When searching for keywords in the index, it might not be important to find the related words if they are not yet known. For instance if you are looking for a relation between t_1 and t_2, it might not be important that t_1 relates to t_3. Therefore, it is desirable to create a multi-layer representation that represents the relation $\mathcal{R}$ between t_1 and t_2 by assigning a relation ID rid_x to it. This creates the following inverted file:

t_1	$\rightarrow$	$(d_y \rightarrow (rid_x/\mathcal{R}))$
t_2	$\rightarrow$	$(d_y \rightarrow (rid_x/\mathcal{R}^{-1}))$

A representation using relation IDs occupies less space than using pointers to global terms. Relation IDs are unique within the granularity of the layer, which in the example just described are documents (i.e. they are unique within one document). For this representation, the retrieval of one term does not cause the retrieval of the other, but once they are both known, they can be linked together.

For instance, the subject relation between *John* and *writes* in the sentence "*John writes in his notebook*" can be expressed by the following structure:

John	$\rightarrow$	$(3100 \rightarrow (15/subj))$
writes	$\rightarrow$	$(3100 \rightarrow (15/subj^{-1}))$

Where 3100 is the document ID, and 15 is the relation ID within the document. In this example IDs were arbitrarily chosen, where in practice they are assigned sequentially starting at 0.

We can also include some extra information about the relations such as the role that each term plays in it. For instance, consider that we have a relation of type $\mathcal{R}$ among three terms (t_1, t_2 and t_3) and each plays a specific

role in the relation, $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ respectively. We can represent this relation by

t_1	$\rightarrow$	$(d_y \rightarrow (rid_x/\mathcal{R}/\mathcal{A}))$
t_2	$\rightarrow$	$(d_y \rightarrow (rid_x/\mathcal{R}/\mathcal{B}))$
t_3	$\rightarrow$	$(d_y \rightarrow (rid_x/\mathcal{R}/\mathcal{C}))$

For instance, the sentence "*This fruity Riesling wine has a golden colour with peach and honey bouquet*" represents the tasting notes that highlight a few characteristics of a Riesling wine. The *taste* relation between a *wine* and its description consists of *appearance*, *flavour* and *aroma*, which can be represented with the structure:

fruity	$\rightarrow$	$(1505 \rightarrow (20/taste/flavour))$
Riesling	$\rightarrow$	$(1505 \rightarrow (20/taste/product))$
golden	$\rightarrow$	$(1505 \rightarrow (20/taste/appearance))$
peach	$\rightarrow$	$(1505 \rightarrow (20/taste/aroma))$
honey	$\rightarrow$	$(1505 \rightarrow (20/taste/aroma))$

Where 1505 and 20 correspond to arbitrarily chosen document and relation IDs.

Our choice of representation is similar to the study proposed by Fuhr and Gövert [43], which we described in Section 2.4.3.1. The similarities include the encoding of structural information in the posting list, the use of a data prefix to indicate the type of information that has been added, and the removal of redundant information, which reduces the size of the posting list. However, Fuhr and Gövert's method is used to record paths in XML trees, whereas our work takes a less structural view of the problem and defines it as a way to encode linguistic information. Fuhr and Gövert method encodes different XML structures, while our work is optimised to work with fixed structures specified by linguistic relations. Thus, our method does not need to encode and decode path information, which means that we can achieve higher inverted file compression, and better indexing and retrieval speeds.

4.3.1 Layer Representation

We use one single file to encode the information of the multi-layer inverted file representation. Therefore, every piece of information that is recorded in

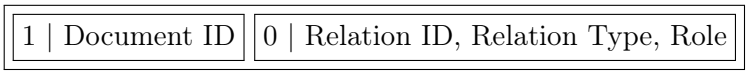


Figure 4.1: The representation of layers types is differentiated by the use of byte prefixes

the inverted file is type defined by a data prefix. This prefix consists of a number of bits capable of differentiating the number of layer types used. For instance, as shown in Figure 4.1, if there are two different layer types used, one layer will include the prefix 0 while the other will have the prefix 1.

Layers can be used to represent single data types such as integer numbers for document IDs or more complex data structures, involving a combination of data types. The layers shown in Figure 4.1 illustrate this concept; the first layer contains a single integer, and the second layer contains three different data types. The types of data that are to be represented in a layer are defined by the implementation of the IR system using the multi-layer framework.

It is important to point out that layers are used to distinguish data types that either are contained in each other or are independent of each other, meaning that if two types of information are always to appear alongside, then they should form a layer by themselves and not be two different layers. This helps to avoid unnecessary prefix overhead that would exist if all information was declared as separate layers. For instance, if we want to include the frequency of a term in a document, we should include that information on the same layer as the document ID, instead of creating two different layers for them. On the other hand, positional information should form a different layer than the document ID layer, since all positions are contained within documents.

Because of the different types of data and amounts of information represented at each layer, it is important to have a mechanism to detail the size of data represented at each layer. This is done by detailing the amount of trailing bytes after the data type prefix. Figure 4.2 illustrates this information using two different types of data that occupy a different number of bytes. The number of bytes for each particular data structure is the amount

Figure 4.2: Representation of two layers types, one representing Document ID using 3 trailing bytes (binary 11) and the other with relation information occupying two bytes (binary 10)

Figure 4.3: Dependent layers such as sections in a document must appear after the document is defined

of trailing bytes plus the remaining bits of the first byte, discounting the prefix bits, e.g. if the date type and size prefixes occupy 4 bits, a type with 3 trailing bytes will occupy 28 bits. By describing the size of data, it is possible to obtain a faster access to the structure by implementing a simple compression technique, where the data would only occupy the minimum amount of bytes. For instance, with a 4-bit prefix, document IDs from 0 to $2^4 - 1$ will be represented using only 1 byte, while document IDs from 2^4 to $2^{12} - 1$ will need 2 bytes, and so forth up to $2^{28} - 1$. Other techniques for compression of the inverted files such as variable byte coding could also be used [86, Section 5.3]. However, as shown by Trotman [150] the advantage of using advanced compression techniques is not always clear since compression techniques may sometimes worsen the performance of a system.

Each individual layer can represent a self-contained and independent piece of information or can be part of an outer layer. The relation between layers is not part of the inverted file representation but it affects the way it is represented. If two layers are dependent on each other, the outer layer must appear before the inner layer. In the example in Figure 4.3, if a section layer is part of a document layer, all information presented regarding the section layer is part of and must appear after the information of the relevant document layer. If three different sections are part of the same document, as shown in Figure 4.3, only one document representation is defined, followed by three section representations.

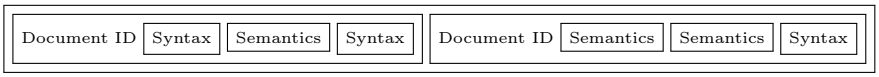


Figure 4.4: Layers that do not depend on each other can appear in any order

However, if two layers are independent of each other, the data of both layer types have no particular order of appearance in the inverted file. This can be observed in Figure 4.4 since the layers for syntax and semantics do not necessarily have an order of appearance as neither depends on each other. It can also be observed that both syntax and semantics are part of a document; therefore, they must appear after the definition of the document layer.

4.3.2 Representing Different Index Types

The multi-layer approach allows different information to be represented in the inverted file of an IR system. Because our work focuses on QA, we tested our framework with linguistic features which included n-grams, syntactic relations and semantic roles. We propose the usage of linguistic indices separately; however, they can be easily combined if required.

Bag-Of-Words

The BoW approach is the simplest implementation possible. For BoW, the terms are linked to a set of documents and their frequency of occurrence:

$$\boxed{(t_i/df_i)} \rightarrow \boxed{(d_1/tf_1,\ d_2/tf_2, \ldots, d_k/tf_k)}$$

The position information could also be added if passage retrieval or any proximity ranking function is to be used:

$$\boxed{(t_i/df_i)} \rightarrow \boxed{(d_1/tf_1 \rightarrow (p_1, p_2, \ldots, p_x), \ldots, d_k/tf_k \rightarrow (p_1, \ldots, p_y))}$$

Notice that in the above representation df_i is equal to k, tf_1 is equal to x, and tf_k is equal to y.

n-grams

n-grams can be calculated with the addition of positional information, as shown in the previous representation, by associating terms in the inverted

file, and by creating a relation ID for each n-gram. The addition of positional information is the preferred method for n-gram retrieval since it does not constrain the size of the n-gram (which can be defined at retrieval time). However, the use of positional information can be more costly for retrieval than presenting related terms or relation IDs. As an example, the following representation demonstrates how trigrams built from a sequence of terms $\{t_1, t_2, t_3, t_4, t_5\}$ can be added to the index when the associated terms are presented:

t_1	$\rightarrow$	$(d_y \rightarrow (p_1/t_2/t_3))$
t_2	$\rightarrow$	$(d_y \rightarrow (p_2/t_1/t_3,\ p_1/t_3/t_4))$
t_3	$\rightarrow$	$(d_y \rightarrow (p_3/t_1/t_2,\ p_2/t_2/t_4,\ p_1/t_4/t_5))$
t_4	$\rightarrow$	$(d_y \rightarrow (p_3/t_2/t_3,\ p_2/t_3/t_5))$
t_5	$\rightarrow$	$(d_y \rightarrow (p_3/t_3/t_4))$

As observed in the example, the inverted file will contain some redundancy, which causes the size of the index to increase and therefore decrease overall efficiency. Another way to represent n-grams includes associating relation IDs for all n-gram terms. This strategy is similar to the way of encoding relations that was previously described without including the type of the relation.

t_1	$\rightarrow$	$(d_y \rightarrow (p_1/rid_1))$
t_2	$\rightarrow$	$(d_y \rightarrow (p_2/rid_1,\ p_1/rid_2))$
t_3	$\rightarrow$	$(d_y \rightarrow (p_3/rid_1,\ p_2/rid_2,\ p_1/rid_3))$
t_4	$\rightarrow$	$(d_y \rightarrow (p_3/rid_2,\ p_2/rid_3))$
t_5	$\rightarrow$	$(d_y \rightarrow (p_3/rid_3))$

Syntactic Information

Syntax can be represented by many different formalisms. One such formalism is represented by syntactic dependency relations [89]. The dependency relations establish links between head and modifier pairs. Using the multi-layer inverted file framework, it is possible to represent these relations by associating syntactically-related terms. For instance, the subject and object relations between the terms of the sentence "*Orpheus loves Eurydice*" can be represented as:

Orpheus	→	$(d_y \rightarrow (subject/modifier/loves))$
loves	→	$(d_y \rightarrow (subject/head/Orpheus,\ object/head/Eurydice))$
Eurydice	→	$(d_y \rightarrow (object/modifier/loves))$

Representing links between terms can be demanding since it would require pointers to the dictionary, which is a large data structure and so the pointers would require a considerable amount of bits. Since it is not required to compute terms that are not yet known in a retrieval task, we include a relation ID instead of the related term, which fulfils the retrieval requirements and allows a reduction in the inverted file size. The syntactic relations and the head and modifier indicators are also represented by an ID that occupies the minimum amount of bits required.

Orpheus	→	$(d_y \rightarrow (subject/modifier/rid_1))$
loves	→	$(d_y \rightarrow (subject/head/rid_1,\ object/head/rid_2))$
Eurydice	→	$(d_y \rightarrow (object/modifier/rid_2))$

Semantic Roles

Semantic Roles are represented differently to syntactic dependency relations because semantic roles associate whole noun phrases to predicates; also the same noun phrase, or just a part of it, can play different roles for different predicates in the same sentence. For instance, the sentence "*Orpheus wore the costume that Eurydice made*" has two main predicates *wore* and *made*. The phrase "*the costume that Eurydice made*" is the object of *wore* while "*the costume*" is the object of *made*:

$[_{\text{agent}}$ Orpheus $][_{\text{predicate}}$ wore $][_{\text{object}}$ the costume that Eurydice made $]$

Orpheus wore $[_{\text{object}}$ the costume $]$ that $[_{\text{agent}}$ Eurydice $][_{\text{predicate}}$ made $]$

In contrast to syntactic dependency relations, the semantic roles of a sentence need to be represented as a graph, as it is not always possible to draw a semantic role representation as a tree. Take for instance the following sentence from PropBank [107]:

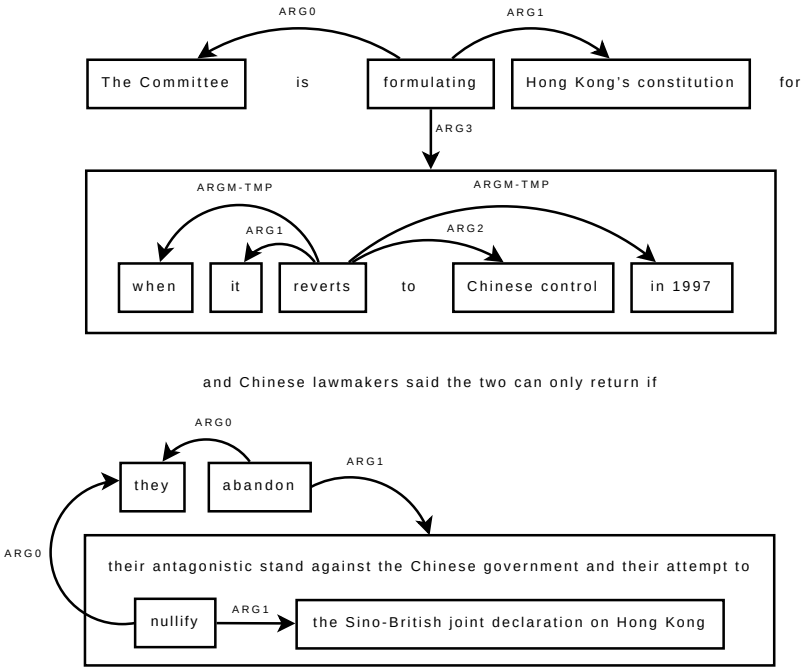


Figure 4.5: PropBank sentence represented as a graph

> *The committee is formulating Hong Kong's constitution for when it reverts to Chinese control in 1997, and Chinese lawmakers said the two can only return if they "abandon their antagonistic stand against the Chinese government and their attempt to nullify the Sino-British joint declaration on Hong Kong".*

The annotation of the predicates of this sentence creates a graph structure as shown in Figure 4.5.

In order to include these relationships in the multi-layer inverted file, we adopt a predicate ID pid_k for every predicate and argument relationship. We also include a positional number to be able to locate a specific word in a multi-word noun phrase. Using this representation the sentence "*Orpheus wore the costume that Eurydice made*" is represented as follows.

Orpheus	→	$(d_y \rightarrow (agent/pos_1/pid_1))$
wore	→	$(d_y \rightarrow (predicate/pos_1/pid_1))$
costume	→	$(d_y \rightarrow (object/pos_2/pid_1,\ object/pos_2/pid_2))$
Eurydice	→	$(d_y \rightarrow (object/pos_4/pid_1,\ agent/pos_1/pid_2))$
made	→	$(d_y \rightarrow (object/pos_5/pid_1,\ predicate/pos_1/pid_2))$

4.4 Using the Multi-layer Index

The indexing of individual terms with the multi-layer representation for relational information allows the retrieval of relations even when the other parts of the relations are not known. Therefore, it is possible to retrieve all documents where two terms are related via a specific relation even though a third term is also part of the same relation but not included in the query. This is particularly important for QA; for instance, it is possible to retrieve all documents where *Orpheus* is said to *love* someone, even though we do not know whom he loves. Many strategies for representing term relations, for instance those that concatenate terms and relations, make it hard to match relations when not all involved parts are known. For example, if a relation such as *Orpheus/loves/Eurydice* is represented as a concatenated string, it is hard to extract documents that contain the partial information that Eurydice is loved by someone, or that Orpheus and Eurydice relate to each other in any way.

If we were to use a concatenated string, the partial relations between *Orpheus*, *love*, and *Eurydice* can be extracted if all possible combinations of partial relations are added (where one of the relation component is missing). This means that with this strategy the following terms are created: *Orpheus/loves/Eurydice*, *Orpheus/loves/*, *Orpheus//Eurydice*, */loves/Eurydice*, *Orpheus//*, */loves/*, and *//Eurydice*. This strategy has obvious drawbacks as the dictionary size will become very large and the information is stored with redundancy.

Another way of extracting partial relations is via concatenated strings that include wildcard matching in the dictionary structures. Wildcard match-

Sentence: $[_{\text{arg0}}$ Orpheus$]$ $[_{\text{pred}}$ loves$]$ $[_{\text{arg1}}$ Eurydice$]$

Index:

Orpheus	$\rightarrow$	$(d_{10} \rightarrow (pid_1/arg0/1,\ pid_2/arg0/1),\quad d_{11} \rightarrow (pid_1/arg1/1))$
love	$\rightarrow$	$(d_{10} \rightarrow (pid_1/pred/1,\ pid_3/pred/1),\quad d_{12} \rightarrow (pid_{15}/arg0/1))$
Eurydice	$\rightarrow$	$(d_{10} \rightarrow (pid_1/arg1/1, pid_3/arg1/1))$

Figure 4.6: Example for the retrieval of the relation expressed by the sentence "*Orpheus loves Eurydice*"

ing allows the querying of terms such as *Orpheus/loves/**, *Orpheus/*/Eurydice*, **/loves/Eurydice*, where the position of the wildcard * is replaced by any string. This strategy has the drawbacks of requiring an extra dictionary structure such as a B-tree, to store the terms in the inverse order (to match wildcards at the beginning of the query string), and post-processing techniques when the wildcard is in the middle of the string [86, Chapter 3].

Our strategy facilitates the computation of TF and IDF for any combination of terms and relations. It is possible to compute the occurrence frequency of a full-matching relation tree in a similar way to the frequency of the smaller parts of a relation. The process of computing the TF and IDF of combinations of relational information involves retrieving all information regarding the relations but only considering the ones that fully match the relation requirements. Figure 4.6 shows the retrieval process for the love relation between *Orpheus* and *Eurydice* in the sentence "*Orpheus loves Eurydice*". As observed, there are three relations that involve the search terms in d_10, but only one of them involves *Orpheus* as the argument 0 (arg0) and *Eurydice* as the argument 1 (arg1) of the predicate *love*. As observed in the example, the information that will be used is the one that matches the sought relations.

It is important to allow the relation requirements to be tightened or relaxed according to the retrieval need. This is important for the QA task in cases when one of the terms of the relation is not present in the query and it would likely represent an answer to the question. For instance, when presented with a question such as "*Who defeated Federer in 2008?*" it is

important to retrieve the relations between all the question constituents, but not necessarily to find an exact match on the *Who* term.

The ability of the multi-layer index to retrieve relationships without the constraint of having to include all constituents of the relationship allows the construction of different retrieval ranking strategies. One of the strategies involves the definition of a three-vector ranking based on TF and IDF weights.

4.4.1 Vector Space Ranking

The vector space model [125] represents queries and documents as vectors of words, and defines the similarity of the query vector Q and a document vector D using the cosine of the angle between their vectors. The cosine is calculated, as show in Equation (4.1), from the dot product of both vectors normalised by the product of the vector sizes:

$$sim(\vec{Q}, \vec{D_x}) \equiv \cos\theta = \frac{\vec{Q} \bullet \vec{D_x}}{|\vec{Q}| \times |\vec{D_x}|} \tag{4.1}$$

The dimensions of these vectors are the query terms. The sizes of each dimension are defined by the term weights $W_{i,j}$ where i represents the dimension (query term) and j represents a sequence of terms (which can be a document, a document passage, or a query). In our experiments, $W_{i,j}$ is defined as a traditional TF*IDF function [121], which is the product of TF and IDF as shown in Equation (4.2).

$$W_{i,j} = TF_{i,j} \times IDF_i \tag{4.2}$$

TF and IDF can be defined in many ways; we adopt a well known definition [127], which uses $TF_{i,j}$ as the raw frequency of occurrence of a term t_i in a document d_j, and IDF_i as the logarithm of the division between the number of documents in the collection (n) and the number of documents n_i containing the term t_i, as shown in Equation (4.3).

$$IDF_i = \log \frac{|D|}{|\{d_k : t_i \in d_k\}|} \tag{4.3}$$

4.4.2 Triple-Vector Space Ranking

When using linguistic information, the dimensions of document vectors is not defined by the document terms as it is in standard BoW retrieval. For instance, when syntactic dependency relations are used, the dimensions can be defined as head and modifier pairs. This strategy creates a high precision ranking function since relations are more restrictive than words. However, it degrades recall since many relevant documents do not contain the desired relations. In order to solve this problem, we created a ranking function that combines the similarity function of vectors of different dimensions, including the standard BoW vectors. This ranking function gives high scores to documents that contain the expected relations, which boosts the systems precision, while still considering documents that only contain the query terms, which keeps the overall recall. Thus, our ranking function allows us to improve precision without degrading recall. This will be further demonstrated in the evaluations described in Chapter 6.

The multi-layer inverted-file representation allows us to store each term and their relationships individually, creating a link between unique relations through the relation identifier. This allows the terms to be retrieved individually and to be linked in a relationship if required. We use this framework to implement a vector space retrieval method that uses three distinct vectors: BoW vector (V1), Word-Roles vector (V2), and Two Words Relation vector (V3).

The vector V1 uses words as its dimension; it contains information about every term in the queries regardless of whether they exist in any relation among each other or with other terms. This vector contains the term weights over the entire collection of documents, and it can be used to rank documents in the same way as would a standard BoW index with only document information.

Dimensions		Who	founded	Public	Citizen	$\vec{Q} \bullet \vec{D_x}$	$sim(Q, D_x)$
	IDF	**1.00**	**2.00**	**2.00**	**4.00**		
	Q	0.25	0.25	0.25	0.25	1.5625	1.0000
	D_1	0.00	0.01	0.01	0.01	0.0600	0.9798
TF	D_2	0.01	0.00	0.02	0.00	0.0225	0.4366
	D_3	0.00	0.00	0.03	0.02	0.1100	0.8800
	D_4	0.00	0.01	0.00	0.05	0.2100	0.8358

Figure 4.7: Representation of V1 for question "*Who founded Public Citizen?*"

Dimensions		Who	founded	Public	Citizen	$\vec{Q} \bullet \vec{D_x}$	$sim(Q, D_x)$
		$\overline{\text{subj}}$	$\overline{\text{main}}$	$\overline{\text{attr}}$	$\overline{\text{obj}}$		
	IDF	**2.00**	**3.00**	**2.50**	**4.50**		
	Q	0.25	0.25	0.25	0.25	2.4688	1.0000
	D_1	0.00	0.01	0.01	0.01	0.0888	0.9480
TF	D_2	0.01	0.00	0.01	0.00	0.0256	0.5094
	D_3	0.00	0.00	0.02	0.01	0.0819	0.7746
	D_4	0.00	0.01	0.00	0.02	0.1238	0.8302

Figure 4.8: Representation of V2 for question "*Who founded Public Citizen?*"

Figure 4.7 shows the representation of V1 for the question "*Who founded Public Citizen?*" and four documents. For the sake of simplicity and better comprehension, we have chosen examples with small number of words, and defined the number of dimensions for all documents as 100. For the same reasons, we have also arbitrarily defined the values of TF and IDF.

The vector V2 contains all terms that are found in the query and play the same role in the document collection and in the query. The size of V2 is smaller than V1 because many terms in V1 are not to be included when they do not play the same role. For instance a query "*poker face*" would not return a document containing "*John faced Mark in a game of poker*" when using syntactic relations, since both *poker* and *face* play different syntactic roles in query and document. For this query, *poker* would only be included in V2 when found as an adjunct and *face* as a noun. Therefore, the dimensions of V2 are defined by terms and their roles in a relationship. Using syntactic dependency relations for the question "*Who founded Public Citizen?*", we would obtain the representation in Figure 4.8.

The vector V3 is more restrictive than both V1 and V2. V3 includes

Dimensions		Who $\overline{\underline{\text{subj}}}$ founded	Public $\overline{\underline{\text{attr}}}$ Citizen	Citizen $\overline{\underline{\text{obj}}}$ founded	$\vec{Q} \bullet \vec{D_x}$	$sim(Q, D_x)$
	IDF	**5.50**	**7.50**	**10.00**		
	Q	0.33	0.33	0.33	20.7222	1.0000
	D_1	0.00	0.01	0.01	0.5208	0.9153
TF	D_2	0.00	0.00	0.00	0.0000	0.0000
	D_3	0.00	0.01	0.00	0.1875	0.5492
	D_4	0.00	0.00	0.01	0.3333	0.7323

Figure 4.9: Representation of V3 for question "*Who founded Public Citizen?*"

only those relations that exist in the queries and in the documents at the same time. For instance, V3 will not include documents that contain the relation between the noun adjunct *poker* and another noun such as *cards* in *poker cards*. Therefore, the contents of V3 have high precision and indicate the existence of the terms in fewer documents than previous vectors. The dimensions of V3 are defined by the frequency weights of the occurrence of the relationship between two terms. Using syntactic dependency relations the question "*Who founded Public Citizen?*" would create the vectors shown in Figure 4.9.

The weights corresponding to the TF and the IDF for V2 and V3 are computed at retrieval time. This is particularly important for V3 since it involves more than one term. Because of the way the multi-layer index is organised, IDF is computed when different terms share the same relation. The relations of a query are likely to be at least as rare as their most distinctive terms. Therefore, the dimensions of V2 and V3 will count highly in the overall ranking, in a way that documents that share distinctive relations with queries will have their rankings boosted and will appear at the top of the retrieval list. This ranking strategy will favour documents that contain expected relations; for instance, a document that contains the relation between *Public* and *Citizen* in the question "*Who founded Public Citizen?*" will have a higher ranking than a document where both terms appear with no connection among each other. Even though this is a retrieval technique that favours precision, it does not harm recall since documents containing both

	$sim(\vec{Q}_{V1}, \vec{D}_{V1})$	$sim(\vec{Q}_{V2}, \vec{D}_{V2})$	$sim(\vec{Q}_{V3}, \vec{D}_{V3})$	$sim(\vec{Q}, \vec{D})$
Q	1.0000	1.0000	1.0000	1.0000
D_1	0.9798	0.9480	0.9153	0.9477
D_2	0.4366	0.5094	0.0000	0.3153
D_3	0.8800	0.7746	0.5492	0.7346
D_4	0.8358	0.8302	0.7323	0.7994

Figure 4.10: Final similarity scores for documents in the question example "*Who founded Public Citizen?*"

terms will still be ranked via V1.

The number of relations represented in V3 for every document and the number of documents that will match these relations will be much smaller than V2, which will already contain fewer terms per document than V1. The number of terms in V2 and V3 are compensated by higher IDF values for each individual dimension, which reflect the higher dot product between the document vectors and the query vector. However, because similarity is represented by the cosine difference between query and document vectors, they are bound to the 0 to 1 range for V1, V2 and V3. Because we want to increase the ranking of documents that rank high in V3 and in V2, without penalising documents from V1 (increase precision without decreasing recall), we decided that all similarity rankings for all vector space rankings have equal weights. In this way V1, V2 and V3 will contribute to exactly one third of the final similarity ranking. Therefore, the final similarity functions $sim(\vec{Q}, \vec{D_x})$ is defined as in Equation (4.4):

$$sim(\vec{Q}, \vec{D}) = \frac{1}{3} \sum_{k=1,2,3} sim(\vec{Q}_{Vk}, \vec{D}_{Vk}) \tag{4.4}$$

For the example of the question "*Who founded Public Citizen?*", the final similarity rankings would be as shown in Figure 4.10.

4.5 Performance Evaluation

Our multi-layer inverted file has shown to be capable of representing different types of linguistic information so that this information is readily available

to an IR system. To show the structure usability in IR, we performed some evaluation in memory usage, indexing time and retrieval time.

4.5.1 Disk Usage

Data structures that store all the information required in later stages, even if it is redundant, are likely to allow the process to perform its task very quickly. At the same time, very compact data structures are likely to require more processing to solve a task. This is the normal trade-off between speed and memory consumption in computer science (see an example in [43]). However, in the IR field this trade-off is only true to a certain point. IR structures, such as the inverted file, are already designed for efficiency, and because the amount of data involved in IR is likely to be very large and unlikely to fit into the computer's main memory, the IR process requires the data structures to be kept small so that processes can work in main memory and do not access the disk storage excessively. Even though inverted file compression techniques increase the algorithm complexity, they can improve overall IR speed performance because efficient compression methods are likely to have higher throughput than uncompressed data when considering the cost of disk access and its transfer to main memory [86, page 69]. However, Trotman [150] has shown that compression ratio alone should not be the only factor for choosing a compression technique.

Because the multi-layer inverted file strategy requires prefix bits to signal the layer number and the number of trailing bytes, simple single-layer inverted file structures would not benefit from the strategy. The extra prefix in this strategy would represent an unnecessary overhead. Using 2-bit prefix for the layer number and 2-bit for trailing bytes, the overhead will be in between 50% (single-layer, no trailing bytes) and 12.5% (four-layers, all with three trailing bytes). The multi-layer strategy has the drawback of a high overhead, but it does not have redundant information. Table 4.1 shows a comparison between the representation of different layers in the multi-layer strategy and using uncompressed postings in a single inverted file.

	Overhead	Redundancy	Database Size
Multi-layer	85.7 MB (12.5%)	-	*685.7 MB*
Uncompressed postings	-	98.1 MB (14%)	698.1 MB

Table 4.1: Storage comparison of 600 MB of data between the multi-layer inverted file strategy and several layers represented by their own inverted file

Table 4.1 compares the memory usage for the indexing of a document and term positions using two separate layers in the multi-layer index, and using full postings where the information is always encoded with the document ID and position. For this comparison, we assume that a term occurs on average 1.39 times in every document; this has shown to be the ratio for the AQUAINT corpus, where the average number of posts per term is 138.26 and the average number of unique documents per term is 99.39. The multi-layer is divided into 4 bits for prefix and 28 bits for data (even though in practice only 3 prefix bits would be required). Since we want to compare the difference between overhead and redundancy, the full posting technique uses 28 bits to store each information (56 bits or 14 bytes in total). The full posting technique requires each position to be recorded with a document ID, which represents redundant information that, as shown in Table 4.1, can be higher than the overhead from the multi-layer techniques.

In practice, the difference can be quite significant between the amount of persistent memory used by the multi-layer index and Flint, a state of the art database used by the Xapian Retrieval tool[3]. Table 4.2 compares the disk usage by the Flint database needed to index all non-stopwords of the first 41,116 documents from the NYT part of the AQUAINT corpus with positional information and the same information using the multi-layer index. The documents occupy 297 MB of disk space of which 138.64 MB consists of textual data (disregarding spaces and punctuations); from this amount, 91.38 MB are non-stopwords.

One of the advantages of our strategy over off-the-shelf IR tools is that the multi-layer index has the capability of representing textual relations using a

[3] `http://www.xapian.org`

	List of Terms	Postings	Other	Total
Multi-layer	124 MB	280 MB	47 MB	*451 MB*
Flint	192 MB	579 MB	8 MB	778 MB

Table 4.2: Storage comparison between the multi-layer inverted file strategy and the Flint database from Xapian using the first 41,116 documents from the NYT part of the AQUAINT corpus

#	Text	Baseform	Syntactic rel.	Syntax and morphology
1	Orpheus	orpheu	subj:>2	@SUBJ %NH Heur N NOM PL
2	loves	love	main:>0	@+FMAINV %VA V PRES SG3
3	Eurydice	eurydice	obj:>2	@OBJ %NH Heur N NOM SG
4	.	.		
5	<s>	<s>		

Figure 4.11: Representation of the sentence "*Orpheus loves Eurydice*" when syntactically parsed using the Connexor dependency parser

normal BoW term list. This is important because the term list needs to be kept small for efficiency reasons, since it is normally kept in main memory. When using off-the-shelf IR tools, researchers have used term concatenation to add extra information, which have thus increased the size of the term list and decreased indexing and retrieval performance.

We have compared the indexing of syntactic relations using term concatenation with the Flint database from the Xapian IR tool, an IR tool that uses a MySQL database, and the multi-layer inverted file. Table 4.3 shows the amount of space needed to index all of the articles from 1996 in the XIE portion of the AQUAINT corpus, totalling 93,458 documents and 917 MB of uncompressed syntactically parsed data. An example of a syntactically parsed sentence is given in Figure 4.11. We can observe that the database size for the multi-layer index is many times smaller than the size of other strategies. Among the reasons for this is that the multi-layer strategy encodes syntactic relations by relation IDs, and stores them in the same way as the BoW approach does for positional information, which virtually allows the BoW index and the syntactic relation index to occupy the same amount of disk space (see Table 4.2).

	List of Terms	Postings	Other	Total
Multi-layer	128 MB	280 MB	46 MB	*454 MB*
Flint	422 MB	1.24 GB	6.7 MB	1.66 GB
MySQL	8.5 MB	1.06 GB	10.52 MB	1.08 GB

Table 4.3: Storage comparison between the multi-layer inverted file strategy, the Flint database from Xapian, and using MySQL for the indexing of 93,458 documents of the AQUAINT corpus

The Flint database occupies a large amount of disk space for the term list, which in the syntactic relation case, holds the concatenated version of every syntactic pair with their positional information. For instance, the sentence "*Orpheus loves Eurydice*" would contain two positions. Position 1 would contain the relations ($Orpheus > subj$), ($Orpheus + love$), and ($love < subj$); and position 2 would contain the relations ($Eurydice > obj$), ($Eurydice + love$), and ($love < obj$). This causes the number of terms to be very large as can be observed in Table 4.3.

For the MySQL approach, because of the relational database capability of storing relational data, there is no need to build the syntactic relations in the term list, which showed to be quite optimised. The term list occupies 8.5MB, which is a fraction of the memory usage of the other techniques. However, the posting list can be quite large due to the need for representing relations by reproducing all combinations of term IDs and relation IDs.

More details on the implementation of the multi-layer index are shown in Appendix A, but it is important to note that the multi-layer implementation does not use any compression techniques beside the focus of not representing redundant information; this reduces the index size but may compromise the speed.

4.5.2 Speed Performance

The large amount of data that needs to be indexed implies that the indexing process of an IR system must be fast enough to handle it. There is also an expectation that the system response to a query should be fast. Because of

	Time (s)	Documents/Seconds
Multi-layer	378.03	247.22
Xapian	*338.22*	*276.32*

Table 4.4: Comparison of BoW indexing times between the multi-layer inverted file strategy and the Xapian IR tool using 93,458 documents

this, speed performance is one of the main concerns of IR. It is important to have a system that can produce reliable results and still perform its function in a timely manner. Even though we are not necessarily focusing on the speed of the indexing and retrieval components at this stage, the multi-layer indexing strategy has achieved this goal by allowing a fast indexing and retrieval of information.

We have observed that the speed for the multi-layer approach is slower than state-of-the-art retrieval systems for BoW retrieval. Table 4.4 shows a comparison between the indexing speeds of the multi-layer approach and the Xapian system. The machine used for all experiments in this chapter is an Intel(R) Pentium(R) 4 HT 2.80GHz with 2.0 GB RAM. We can observe that the Xapian system outperforms our approach for BoW indexing. However, the same small difference of performance is observed in favour of the multi-layer technique when comparing with the Xapian system for the indexing of syntactic dependency relations (Table 4.5). Even though the time measurement does not include parsing, which is performed offline, there is a significant speed difference between the BoW indexing and the syntactic dependency relations. This difference is due to the extra parsing time to extract the dependency relations (which is more demanding than the BoW tokenisation), and in the case of the Xapian IR tool the large vocabulary size caused it to be slower than our system. The MySQL approach is by far the slowest indexing approach, performing at least four times slower than the other techniques.

The purpose of an IR system is to obtain lists of documents for user queries; therefore, the architecture of most index databases focuses on improving the speed of retrieval. Our system, on the other hand, is designed

	Time (s)	Documents/Seconds
Multi-layer	*796.73*	*117.30*
Xapian	890.42	104.96
MySQL	3,531.10	26.47

Table 4.5: Comparison of the indexing times of 93,458 syntactically parsed documents

	Time (s)	Queries/Seconds
Multi-layer	14.30	24.55
Xapian	*4.77*	*73.58*

Table 4.6: Retrieval time of 351 BoW queries of the multi-layer inverted file strategy and the Xapian IR tool

to provide a fast-enough access to the retrieval of relational information, but not necessarily to provide the fastest access possible.

Table 4.6 shows a comparison between the retrieval speeds of the multi-layer approach and the Xapian system using the indices of Table 4.4 with TREC Questions from the 2004 QA track. The comparison shows that Xapian is able to retrieve queries many times faster than our approach. This does not invalidate our approach since the multi-layer retrieval speed of 24.55 queries per second is a rate that allows us to perform comparative experiments using a large amount of questions in a short time. For example, we can retrieve the 1448 questions from the TREC 2004, 2005, and 2006 in less than one minute.

For the syntactically parsed documents, the performance of the retrieval component degrades significantly. As we can observe in Table 4.7, the performance of all techniques is at least six times slower than the BoW. However, on the contrary to the BoW results the multi-layer approach is significantly faster than Xapian. Similar to the indexing performance, MySQL is the worst performance technique even though the MySQL system is developed with extensive use of term caching and offline indexing of tables[4] to improve

[4]Most of the table indexing was done after the IR indexing phase but before the IR retrieval phase.

#	Text	Baseform	Syntactic rel.	Syntax and morphology
1	Who	who	subj:>2	@SUBJ %NH <Interr> PRON WH NOM
2	loves	love	main:>0	@+FMAINV %VA V PRES SG3
3	Eurydice	eurydice	obj:>2	@OBJ %NH Heur N NOM SG
4	?	?		
5	<p>	<p>		

Figure 4.12: Representation of the question "*Who loves Eurydice?*" when syntactically parsed using the Connexor dependency parser

	Time (s)	Queries/Seconds
Multi-layer	*92*	*3.82*
Xapian	154	2.28
MySQL	167	2.10

Table 4.7: Retrieval time of 351 queries for syntactically parsed documents

retrieval performance over standard query intensive MySQL retrieval. An example of a syntactically parsed query is given in Figure 4.12.

4.6 Concluding Remarks

In this chapter, we presented an inverted file structure that incorporates different layers in order to represent different types of information. We presented different ways that the multi-layer representation can be used to represent different types of linguistic information, and we have proposed a triple vector space based ranking strategy for retrieving relations using the proposed structure. Finally, we showed the feasibility of the multi-layer indexing techniques by measuring the speed of indexing and retrieving as well as the size of the database it generates. We have shown that the multi-layer approach uses less disk space and performs faster for both indexing and retrieval of relational information than a state-of-the-art BoW retrieval tool.

5

Question Prediction Language Model

It has long been discussed that semantic models are able to help in tasks such as QA and IR, however, few studies have made use of such models; some exceptions include [66, 102, 134, 141]. A common problem with these studies is the poor coverage of the semantic resources, causing the techniques to be of limited utility [134].

This chapter proposes the use of a language representation that specifies the relationship between the terms of a sentence using question words. The proposed representation is tailored to help the search for documents that contain an answer to a natural language question.

We describe a language model that focuses on extracting a simple semantic representation of English texts that can be easily stored in digital databases and processed by IR tools. We focus on extracting a particular kind of information that helps to find the location of a text that has some likelihood of answering a specific kind of question. The model and its semantics are defined as the Question Prediction Language Model (QPLM).

This chapter is divided into four sections. Section 5.1 presents the Question Prediction Language Model, some of its features and a comparison with other semantic annotation models. Section 5.2 proposes some usages for the QPLM model and Section 5.3 describes how the model is built, how it is applied to text and discusses some performance evaluations. Section 5.4 completes the chapter with some concluding remarks and future work.

5.1 Model Definition

QPLM represents sentences by specifying the semantic relationship among its words using question words. In this way, we focus on dividing the problem of representing a large sentence into a number of small questions that could be asked about its components.

Each small question is represented as a triple as specified in Equation (5.1):

$$\langle \mathcal{Q}(\mathcal{W}) \rightarrow \mathcal{A} \rangle \tag{5.1}$$

Where $\mathcal{Q}$ is a question word, $\mathcal{W}$ is a word concerning $\mathcal{Q}$, and $\mathcal{A}$ is a word answering the relation between $\mathcal{Q}$ and $\mathcal{W}$.

The relation $\mathcal{Q}$ is represented as a question word such as *Who* or *Where*, but more importantly, it should represent what question can be asked about term $\mathcal{W}$ that the term $\mathcal{A}$ would suffice as an answer. QPLM represents the relationship among keywords of a sentence as short questions in a way that a sentence such as "*John eats ham*" is represented by the following two triples: $\langle Who(eat) \rightarrow John \rangle$ and $\langle What(eat) \rightarrow ham \rangle$. With QPLM, it is possible to answer short questions that focus on events in between the context of a sentence, such as "*Who eats ham?*" and "*What does John eat?*"

As another example, the relation between *John* and *asked* in the sentence "*John asked for a flag to be placed in every school*" could be understood either as a question about the action that John performed or about the person that has asked something, depending on which word is used as $\mathcal{W}$ or $\mathcal{A}$. The relation between $\mathcal{W}$ and $\mathcal{A}$ can be closely related to syntax such as the subject and object relationships, or it can define a more semantic type of relation such as cause and effect. An example of a cause and effect relation is the one between *died* and *cancer* in the sentence "*John died of liver cancer*". Such relations are defined by the question word *Why* in QPLM.

The usage of question words such as *Who*, *What* and *Why* allows the model to closely resemble natural language questions. It is our intention to maintain the resemblance to real questions so when converting natural

QPLM: $\langle \mathcal{Q}(\mathcal{W}) \rightarrow \mathcal{A} \rangle$	**sentence**
$\langle Who(dance) \rightarrow couple \rangle$:	"*The couple*$^{\mathcal{A}}$ *danced*$^{\mathcal{W}}$ *the samba*"
$\langle Whom(award) \rightarrow Marc \rangle$:	"*The prize was awarded*$^{\mathcal{W}}$ *to Marc*$^{\mathcal{A}}$"
$\langle Where(land) \rightarrow Sydney \rangle$:	"*The aeroplane landed*$^{\mathcal{W}}$ *in Sydney*$^{\mathcal{A}}$"
$\langle FromWhere(depart) \rightarrow Perth \rangle$:	"*They will depart*$^{\mathcal{W}}$ *from Perth*$^{\mathcal{A}}$ *today*"
$\langle ToWhere(go) \rightarrow London \rangle$:	"*They went*$^{\mathcal{W}}$ *to London*$^{\mathcal{A}}$ *yesterday*"
$\langle HowMany(orange) \rightarrow ten \rangle$:	"*He ate ten*$^{\mathcal{A}}$ *oranges*$^{\mathcal{W}}$"
$\langle HowOld(John) \rightarrow eight \rangle$:	"*John*$^{\mathcal{W}}$ *is eight*$^{\mathcal{A}}$ *years old*"
$\langle When(open) \rightarrow Friday \rangle$:	"*The play opens*$^{\mathcal{W}}$ *on Friday*$^{\mathcal{A}}$"
$\langle Why(oversleep) \rightarrow depressed \rangle$:	"*Brad overslept*$^{\mathcal{W}}$ *because he was depressed*$^{\mathcal{A}}$"

Figure 5.1: Examples of sentences and QPLM triples

language questions to QPLM, the part containing the question word will force the relation between the question word and their main verb to be the same as the question word. For instance, because the question "*Who danced?*" has the relation between *Who* and *danced* defined as *Who*, it creates the QPLM triple $\langle Who(dance) \rightarrow Who \rangle$. In this case the term *Who* that occupies the $\mathcal{A}$ position behaves like a wildcard. The advantage of having the question word represented as itself in the QPLM is that it is then possible to look for sentences that answer a question by filling the space occupied by the placeholder question word. If needed, a further constraint can be placed on the type of the answer that fits the placeholder location. For the previous question, the sentence "*The couple danced*" is represented as $\langle Who(dance) \rightarrow couple \rangle$ which is a good match for the question triple $\langle Who(dance) \rightarrow Who \rangle$.

Choosing which question word $\mathcal{Q}$ is used in the relation between $\mathcal{W}$ and $\mathcal{A}$ is done by defining what question is $\mathcal{A}$ answering in relation to $\mathcal{W}$. The types of question are defined depending on the purpose of the task and the corpus used. Some examples of the relations used by QPLM are shown on Figure 5.1.

Some of these relations can be ambiguous and are presented as a hierarchy. This hierarchy is needed when the context provided to the annotation

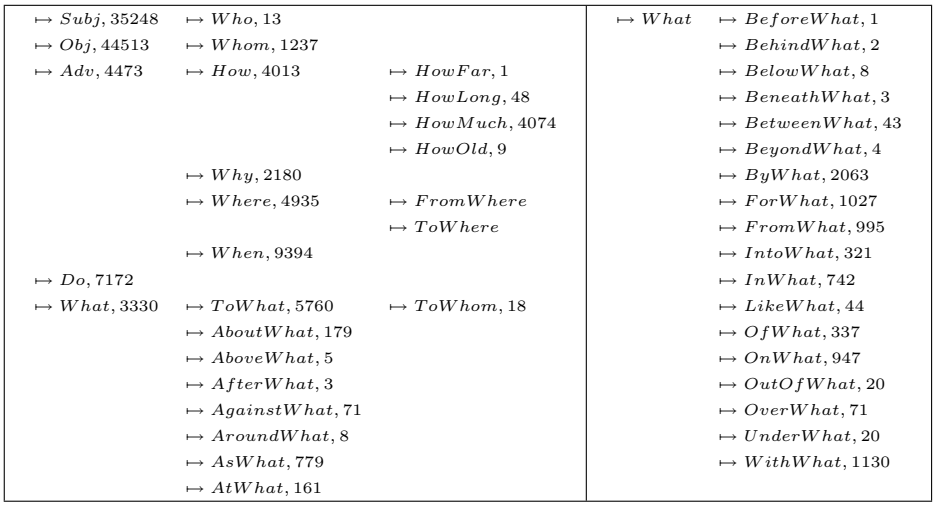


Figure 5.2: Hierarchy of QPLM's question labels. The numbers on the right side represent the number of times the label appears in the QPLM representation of PropBank.

is not sufficient to make a distinction on a more specific label, and when the level of specification does not reflect the desired semantics. For instance, as illustrated in Figure 5.1, we add the prepositions *From* and *To* to the relation *Where*, creating the relations *FromWhere* and *ToWhere*, which solves the direction ambiguity of *Where* when required. For the sentence "*The aeroplane landed in Sydney*", the correct annotation for the relation between *landed* and *Sydney* is the more generic label *Where*, not the more specific *FromWhere* or *ToWhere*. Figure 5.2 presents a manually built hierarchy of questions types that are used with QPLM. All the labels of this hierarchy were chosen when performing the conversion of PropBank to QPLM (this process will be highlighted in Section 5.3.1).

As observed in Figure 5.2, some labels such as *subj*, *obj*, *adv* and *do* are not in the *wh*-question format. One of the reasons is that it would be incorrect to use a label *Who* or *Whom* together with a non-person or organisation; also, many *obj* relations would have a generic *What* label associated to it, which would cause the classification to be very generic. The hierarchy presented in Figure 5.2 is not entirely linguistically motivated. Because some of these

labels can form different sub-branches of the classification, we decided on a simplified model where for example, the distinction between *What* questions are more on the string level than the semantic of the relations.

5.1.1 Directional Semantic Relations

QPLM represents word-to-word relations, focusing on scalability, rather than completeness, as the number of questions for any given literary material can be extremely large. In this format, every relation between two words could be represented as at least two questions. Taking the sentence "*John runs*", we can understand it over two distinct points of view: (1) John does the action of running, and (2) the person who runs is John. Therefore, we can generate two questions about this relationship: (a) "*Who runs?*", and (b) "*What does John do?*". QPLM represents only one of these questions, even though every relation can generate two distinct questions, one from each direction of the relation. When choosing between the two options of question formulation, the main word (head) of the relation should be selected as the $\mathcal{W}$ term. In the previous example QPLM will only create the triple $\langle Who(run) \rightarrow John \rangle$ and *will not* create the triple $\langle WhatDo(John) \rightarrow run \rangle$, because the second triple provides redundant information.

5.1.2 Natural Language Question Generation

The sentence representation created by QPLM works at word level relations. In this way, several triples are formed for every sentence converted to the QPLM representation allowing natural language questions to be generated. The sentence "*Today, John decided to buy Intel stocks on the NASDAQ exchange*" and its QPLM triples, represented in Figure 5.3, can be represented as a directed graph of semantic relations, as shown in Figure 5.4.

Figure 5.4 illustrate that these type of relations are different to syntactic parse trees, and that they are better represented as graphs. We can also observe that when a node is removed from this graph, a full NL question regarding the sentence can be asked using the remainder of the graph and

"*Today, John decided to buy Intel stocks on the NASDAQ exchange*"
$\langle When(decide) \rightarrow today \rangle$
$\langle Who(decide) \rightarrow John \rangle$
$\langle What(decide) \rightarrow buy \rangle$
$\langle Who(buy) \rightarrow John \rangle$
$\langle What(buy) \rightarrow stock \rangle$
$\langle Which(stock) \rightarrow Intel \rangle$
$\langle Where(buy) \rightarrow exchange \rangle$
$\langle Which(exchange) \rightarrow NASDAQ \rangle$

Figure 5.3: QPLM representation

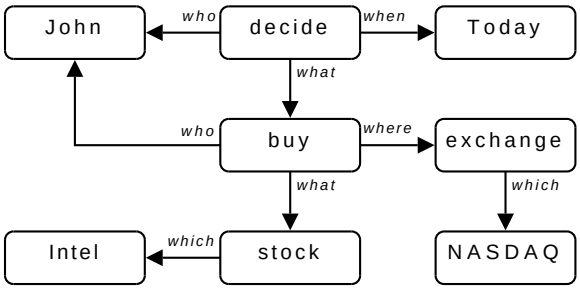


Figure 5.4: Graph representation

answered with the removed node. For instance, by removing the term *John*, we can create the question "*Who decided to buy Intel stocks on the NASDAQ exchange today?*". The same is valid for every node, creating the question set represented in Table 5.1. Observe that the deletion of node *buy* creates the question "*What did John decide?*", which requires an answer centred on the clause of the deleted node: "*to buy Intel stocks on the NASDAQ exchange*".

The language model created using the proposed semantic representation is given the name of Question Prediction Language Model because of its feature of being able to generate full natural language questions from the relations among its triples.

5.1.3 Comparison with Existing Work

QPLM represents a labelled semantic relation between two words in a sen-

Node/Answer	Question
Today	When did John decide to buy Intel stocks on the NASDAQ exchange?
John	Who decided to buy Intel stocks on the NASDAQ exchange today?
decided	What did John do?
buy	What did John decide today?
stock	What did John decide to buy on the NASDAQ exchange today?
Intel	Which stock did John decide to buy on the NASDAQ exchange today?
exchange	Where did John decide to buy Intel stock today?
NASDAQ	On which exchange did John decide to buy Intel stock today?

Table 5.1: Question generation for example sentence

tence via semantic triples. The type of semantic relations represented by QPLM has similarities with existing models, such as the one used by SRL tools [87] and the model proposed by Litkowski [77].

SRL represents the relations between a predicate and their arguments as previously described in Section 2.4.2. PropBank [107] is a manually annotated corpus with semantic roles, which provides the basis for many SRL tools [71, 142]. Even though these tools have proven to be quite accurate (with F1 up to 80% [87]), they demand a computational power that is not available to most research groups when working with medium-to large corpora. In our experience, the annotation of a 3 GB corpus such as the AQUAINT [50], using a SRL tool such as SwiRL from Surdeanu and Turmo [142], can take more than one year using a standard PC configuration.[1] In our experiments with the SwiRL tool, it takes three minutes to parse one AQUAINT article (with more than one million in total), which means we would need six years of machine power to be able to completely parse the AQUAINT corpus.

QPLM was developed as an alternative annotation for semantic relations that can efficiently process large quantities of text. The semantic relations of QPLM assigns relations between words instead of noun phrase-to-predicate relations of PropBank semantic roles. A similar approach using semantic relation triples is taken by Litkowski [77] in his QA system. As described in

[1] Intel(R) Pentium(R) 4 HT 2.80GHz with 2.0 GB RAM

Section 2.4.2.1, Litkowski's semantic relation triples use surrogate terms to represent the semantic roles such as *SUBJ*, *OBJ*, *TIME*, *NUM*, *ADJMOD* and the prepositions that head prepositional phrases.

Litkowski's triples are based on handcrafted heuristics of syntactic clues. QPLM also relies on syntax, however as we will describe in Section 5.3.2, QPLM uses automatically learned rules to generate the semantic relations and it simplifies the relations to a word-to-word level. Differing from Litkowski, QPLM presents a hierarchy of semantic relations that can be more specific for certain cases (e.g. *FromWhere*, *Who*) and more generic in others (e.g. *What*).

SRL and the semantic relation triples from Litkowski have been shown to aid the QA task [77, 78, 79, 80]. However, little has been discussed about speed performance and its usage on IR beyond the answer extraction task of QA. The kind of semantic information used in QPLM is a simplified, but comprehensive, annotation scheme that allows an effective indexing and retrieval of information.

5.2 Using QPLM

QPLM is a way of representing documents using semantic relations between words in a sentence. It involves an uncomplicated structure that allows the representation of a powerful type of semantic relation. This structure and its information allow the usage of such a model in a range of language oriented processing tasks.

QPLM was named after its question generation feature, which takes a sentence and creates a list of simple questions that the sentence is able to answer. Examining the Remedia corpus published by Remedia Corporation and annotated by MITRE Corporation [58], we noticed that many of the questions used could have been generated by QPLM. Figure 5.5 on page 123 shows a document from the Remedia corpus and its questions.

QPLM could be used to automatically create or assist the creation of a reading comprehension task and its evaluation using simple text. However,

1989 Remedia Publications, Comprehension/5W's2

Sailors End Long Trip

(SOUTH SEA ISLANDS, August 1, 1947) Six sailors finished a 5,000-mile sailboat trip yesterday. The men crossed the sea just like people did long ago. Their boat did not have an engine. All they had were oars. The men left Peru on April 28. They wanted to show how, in the past, people traveled miles across the seas. For supplies, they took food and water. They cooked fish on a small stove. For company, they took along a parrot. **The captain's name is Thor**[1]. Since he landed, he has told many tales. Often, he and his crew ran into storms. One time, someone fell overboard. They quickly saved him with ropes. For months they did not see land. At last, they spotted some birds. But more weeks went by without land. **On day 101 of the trip, they reached the shores of the South Sea Islands**[3]. **The name of their boat is the Kon Tiki**[2]. It traveled 4,950 miles. But the men don't plan to sail it back to Peru. They will take a ship home. They are going to let someone else do all that hard work.

1. **Who is the captain of the sailboat?**
2. **What is the name of the boat?**
3. **When did they reach the shore?**
4. Where did the trip start?
5. Why did they make the trip?

Figure 5.5: Example document from the Remedia corpus. Question 1, 2 and 3 could have been generated by QPLM. The sentences that could have been used to generate these questions, and that contain their answers are highlighted and marked with the respective question number.

the simple questions of the Remedia corpus are unlikely to be similar to the ones used in modern QA systems. For this reason, the main proposed usage of QPLM for QA does not include question generation.

The task of finding answers for questions often demands knowledge that spans across several sentences as well as different documents. Even though QPLM works on the sentence level, the additional information that QPLM provides to words alone will likely help to find the answer even among disconnected relations.

5.2.1 A Partial Question Answering System

The proposed usage of QPLM involves the development of an IR technique that resembles a QA system. The resemblance is because QPLM allows more than just the retrieval of information, it also implicitly analyses the question and indicates a possible answer. One could see QPLM as a semi-QA technique, since QPLM by itself will not constitute a full QA system nor would it perform as one.

5.2.1.1 QPLM for Question Analysis

Question Analysis focuses on finding what type of answer is required by a question, as described in Section 2.2.3.1. QPLM represents all sentences, including questions, as small questions regarding the relation between its words. When converting a question to QPLM the question word is treated similarly to the sentence words, causing it to occupy the $\mathcal{A}$ position in the QPLM triple. At the same time, the same question word will be assigned to the $\mathcal{Q}$ position.

The QPLM representation of questions, which assigns the same question word for both the $\mathcal{Q}$ and the $\mathcal{A}$ positions, allows a form of question analysis. This representation informs us that a particular question expects an answer to fill the $\mathcal{A}$ slot occupied by the question word. For instance, the question "*Where did Chavez graduate from?*" is represented by the following triples:

$$\langle \mathit{Where}(\mathit{graduate}) \rightarrow \ \mathbf{Where} \rangle, \langle \mathit{Who}(\mathit{graduate}) \rightarrow \ \mathit{Chavez} \rangle.$$

Question and Answer	QPLM representation
Who killed JFK?	$\langle Who(kill) \rightarrow$ **Who**$\rangle$, $\langle Whom(kill) \rightarrow JFK\rangle$
JFK was killed by Oswald	$\langle Who(kill) \rightarrow$ **Oswald**$\rangle$, $\langle Whom(kill) \rightarrow JFK\rangle$
When was Hitchcock born?	$\langle When(born) \rightarrow$ **When**$\rangle$, $\langle Who(born) \rightarrow Hitchcock\rangle$
Hitchcock was born in 1899	$\langle When(born) \rightarrow$ **1899**$\rangle$, $\langle Who(born) \rightarrow Hitchcock\rangle$

Figure 5.6: Example of questions and matching answers with their QPLM representation

This informs us that we are looking for a *Where* relation having *graduate* as $\mathcal{W}$ and a location as the $\mathcal{A}$ part of the triple.

QPLM does not perform the same type of question analysis as standard question analysers such as the trie classifier presented in Section 3.1. Standard question analysers find the EAT and the question focus, while QPLM finds the type of relation and the place in the triple where the answer is likely to appear. Even though QPLM does not explicitly define the EAT of the question, the EAT can be defined by the type of named entity that is associated with the rule with the question word in the $\mathcal{A}$ position. For instance, the triple $\langle Where(graduate) \rightarrow \ Where^{\mathcal{A}}\rangle$ normally requires the $\mathcal{A}$ position to be filled with a named entity of type *location*.

5.2.1.2 QPLM for Answer Extraction

As the process continues, if a sentence is found to contain a location filling the correct slot of the triple, there is a possibility that the information in $\mathcal{A}$ is the answer. The probability that the answer for a question is the one partially matching the triple with the question word in $\mathcal{A}$ is higher when more triples are matched in proximity to each other. Therefore, it is an indication of an answer if a sentence triple partially matches a question triple containing the question word in the $\mathcal{A}$ position. Figure 5.6 presents some examples of question and answer representation.

The strategy of extracting answers by matching semantic triples is successfully used in a QA system by Litkowski [77]. Even though he uses some very specific categories that could limit the ability to answer questions, it

provides results for as nearly 42% of the questions and had a MRR score of 0.482, meaning that the correct answer is normally presented as the second or third ranked. QLPM can be used by an answer extraction module to deliver the same performance as the aforementioned module, due to the similarity of the semantic triples used by the two works. However, in this thesis we turn our focus away from directly trying to improve the extraction of answers, to how this semantic technique can be applied to the IR stage, so that it can retrieve better quality documents to improve the overall QA results.

5.2.1.3 QPLM in Information Retrieval

IR is one of the main contributors for the QA task. If the IR component performs poorly and the retrieval set does not contain the expected answer for a question, it does not matter how well crafted the other QA components are, the answer will still not be found. Because of this, we focus on using QPLM in IR as a way to improve QA. The use of QPLM is possible by the IR framework presented in Chapter 4, which allows relational information to be incorporated in the search data structures.

The framework includes the relation represented by QPLM in the index, so that the relation can be easily retrieved. The addition of such relations presents a major challenge to the IR tool. QPLM demands that all keywords and relations are indexed, as it requires the ability to fully and partially match a QPLM triple. A full match of triples means that $\mathcal{Q}$, $\mathcal{W}$ and $\mathcal{A}$ will be the same for the search index and query. In a similar way, a partial match occurs when two components of the triple are found to be the same. The focus on partial matching is important since it allows the retrieval of similar triples even when they are not the same and it will permit the indication of a possible answer when the question word occupies the $\mathcal{A}$ position.

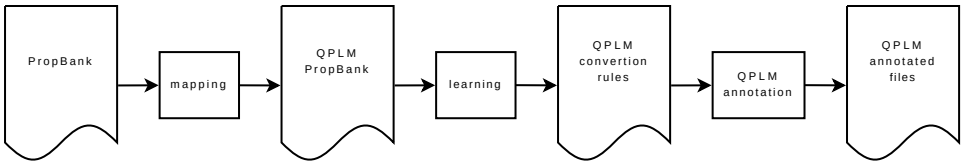


Figure 5.7: Overview of the process of creation and usage of pattern rules

5.3 Building QPLM

Because some of the semantic relations in QPLM correlate with syntax, we perform the semantic evaluation process by applying some mapping rules from syntactic parse trees to our triples. We create the rules automatically in order to obtain a portable and scalable method.

A training set of QPLM triples is created using mapping rules from a corpus of semantic role labels. Using a syntactic parser and a named-entity recogniser with our training set, we are able to learn pattern rules that we further apply in the processing of the AQUAINT corpus. A brief graphical representation of the process described in this section is illustrated in Figure 5.7.

PropBank [107] is a corpus with annotated predicate-argument relations from the same newswire source of information as the Penn Treebank.[2] We used PropBank as our starting point because it comprises the same consistent textual style, and the predicate-argument relations (also referred to as semantic roles) can be mapped to QPLM triples.

We studied the possibility of using SRL tools to perform the semantic annotation; however, our experiments using these tools showed us that they have not yet achieved a reasonable speed performance. As discussed on page 121, SwiRL would take a few years to fully process the AQUAINT corpus. By contrast, our system takes a couple of days if all the necessary information is already at hand; adding the time required for syntactic parsing and named-entity recognition, the total processing period is not longer than two weeks.

[2] `http://www.cis.upenn.edu/~treebank`

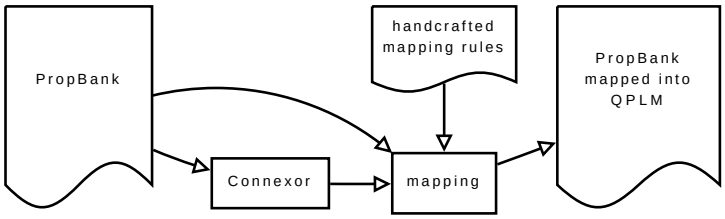


Figure 5.8: Process of mapping PropBank into QPLM

5.3.1 From PropBank to QPLM

PropBank is processed through a set of mapping rules from the predicate-argument relations to QPLM. This process is illustrated in Figure 5.8. We have manually created a set of mapping rules that are applied to the PropBank corpus to convert it to a QPLM format. Because the argument structure of QPLM is made of noun-phrases and QPLM uses only words, we applied a set of heuristics over the syntactic parser to extract the most important word of the argument. These heuristics will be defined shortly in this section.

Using a PropBank map as our training data gives us the benefit of a large training set, but at the same time, it will only create relations that are present in PropBank, thereby excluding some relations that we may wish to include. For instance, relations that do not involve any action are omitted in PropBank, such as the ownership relation in $(\langle Whose(car) \rightarrow \ Maria \rangle)$ and the quantity relation in $(\langle HowMany(country) \rightarrow \ twenty \rangle))$.

Observing how predicates, arguments and function tags relate to each other, we manually created a set of rules mapping from argument-predicate relations to the QPLM. Figure 5.9 shows two examples of mapping rules from PropBank to QPLM, represented in the Python programming language. The rules are represented as embedded Python dictionaries. The dictionary is primarily indexed by argument number which then contains another dictionary structure indexed by functional tags, which then contains a dictionary indexed by predicates. The conversion process will navigate through this structure with the PropBank information until it finds the appropriate

```
SRToQPDict["ARG1']['AROUND']  = {'*':'Where',}
SRToQPDict["ARG2"]['AROUND']  = {'*':'AroundWhat', 'price':'HowMuch', \
                                 'gather':'Where', 'spread':'Where', \
                                 'sprinkle':'Where', 'tangle':'Where'}
```

Figure 5.9: Example of Python mapping rules from PropBank into QPLM

QPLM conversion; if no specific match is found, the most specific wildcard relation (marked with '*') is used. For instance, we convert an *arg1* with function tag *AROUND* to a *Where* relationship, since this relationship always appears with a meaning related to being *around* somewhere. However, not all PropBank relations can be clearly mapped into QPLM: when different QPLM representation were possible, the most generic QPLM relation of them was chosen (i.e. *Where*, instead of *ToWhere*). Another example of rule in Figure 5.9 tells that an *arg2* with function tag *AROUND* in PropBank will be converted into *AroundWhat* unless if involves the predicate *price*, which converts to *HowMuch*, and the predicates *gather*, *spread*, *sprinkle*, and *tangle*, which convert to *Where*.

It is important to stress that if PropBank is extended with additional sentences, the conversion rules will still be applicable to convert this extended corpus. However, if PropBank changes its way of representing semantic role information (argument numbers and functional tags), a new mapping would be required.

As previously specified in Section 5.1.3, the basic differences between both models is that the QPLM triple contains a label representing a more specific semantic relation, and that it associates only the head of the linked phrases. For instance, the sentence "*The retired professor received a lifetime achievement award*" is represented as follows:

(1) **PropBank Semantic Roles**: [$_{\text{arg0}}$ The retired professor] [$_{\text{pred}}$ received] [$_{\text{arg1}}$ a lifetime achievement award].

(2) **QPLM**: $\langle \mathit{Who}(\mathit{receive}) \rightarrow \mathit{professor} \rangle$, $\langle \mathit{What}(\mathit{receive}) \rightarrow \mathit{award} \rangle$

As can be observed in (1), SRL does not provide information about which is the main term of each argument (normally the head of a phrase), while in (2), QPLM represents relations between the phrase heads. In order to find the phrase head, we applied the Connexor syntactic parser [143] to PropBank sentences. However, the phrase heads are not always clearly defined (particularly when the syntactic parse tree is broken due to problems in the parser), which creates an extra difficulty for the mapping process. When a syntactic path cannot be found between predicates and any of the words from the argument, we try to find the head of the phrase by syntactically parsing the phrase by itself. If this also fails to provide us with a head, we simply use the first available non-stopword if possible.

The stage of finding the related phrasal heads showed to be quite important, because not only we would be defining which words relate to each other, but also because if a broken parse tree is found, no rules could be learned from the resulting QPLM triple. An analysis of the data showed us that 68% of the QPLM triples derived from PropBank were generated from an unbroken parse, while the rest used some of the other methods.

Even though our model has similarities with SRL, we are taking a step further in the sense of semantic representation. QPLM has a finer semantic representation, meaning that a predicate argument relation in PropBank might have different representations in QPLM. Our mapping rules takes into consideration not only the number of the argument (e.g., *arg0* and *arg1*) but also the predicate involved and the POS or named entity of the related words. For instance, the representation between the verbs and the subjects of the sentences "*The car travels along the road*" and "*John travels to Hawaii*" are represented by the same argument type in ProbBank style semantic roles (in this case *arg0*). In contrast, QPLM uses different relationships to represent whether the agent of the predicate *travel* is a person or not. Consequently, QPLM builds the following triples $\langle What(travel) \rightarrow car \rangle$ and $\langle Who(travel) \rightarrow John \rangle$.

Even though we cover different aspects of PropBank in our mapping,

we observed that many predicates hold different meanings for the same arguments, which creates a problem for our mapping strategy. This problem is not fixed because of the prohibitive amount of work needed to manually mark all the different meanings for the same predicate in different sentences. In these cases, where the same predicates and the same argument represent different semantics according to the QPLM, we chose the one most representative for the set of sentences using that predicate and argument. For instance, the argument number 3 of predicate *spend* for the majority of the cases represents a quantity of money that is spent (a *HowMuch* label), however we have one case where the argument is *cash* (a *What* label). This type of mapping compromises the accuracy of our conversion, however a randomly selected set of 40 documents was manually evaluated showing that nearly 90% of the QPLM triples are correctly converted. After the mapping is finalised, we obtain a training set of 60,636 rules, including all types of semantic relations from Figure 5.2.

5.3.2 Rule Learning

The PropBank corpus, after being automatically converted to QPLM triples, is used to learn the rules that are used to find the QPLM information of plain text. The QPLM annotation relies on the output of a syntactic parser and of a named-entity recogniser for its annotation and for the rule learning process. We are currently using Connexor [143] for syntax parsing and LingPipe[3] to recognise NEs. Our semantic model uses pattern rules (PRules) created from the representation of the same sentence as syntactic parse trees, MUC style named entity [52], and a list of QPLM triples. PRules describe which QPLM triple can be created from what path of syntactic dependencies with named entities. PRules have a simple human-readable representation that can be manually modified if required. For instance the following PRule defines that if a person is the subject of an active verb, then a *Who* triple is created:

$$\langle \mathit{Who}(\mathcal{W}) \rightarrow \ \mathcal{A} \rangle\text{: } \boxed{\underset{person}{\mathcal{A}}} \xrightarrow{\text{subj}} \boxed{\underset{va}{\mathcal{W}}}$$

[3] http://www.alias-i.com/lingpipe

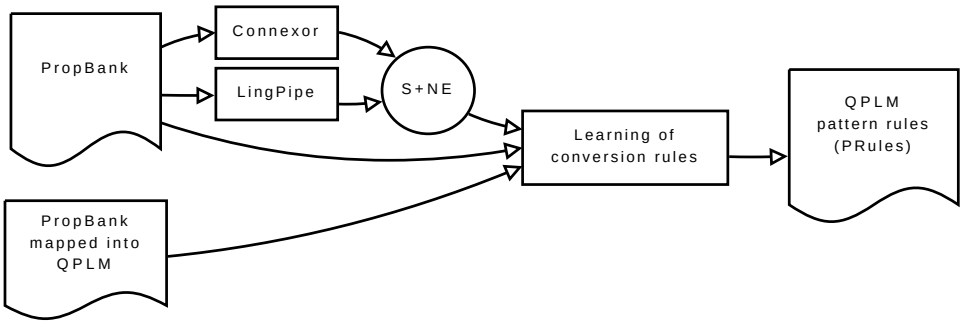


Figure 5.10: Process of learning pattern rules for QPLM

The process illustrated in Figure 5.10 learns a set of PRules using the different types of information shown in Figure 5.11.

1. We use the syntactic parse tree and the named-entity representation, and replace the part of speech of the named entities in the parse tree by their named-entity category. We flatten this syntactic representation by identifying all the leaf-to-root links. These syntactic leaf-to-root paths combined with the named-entity information is defined as S+NE paths. For instance, Figure 5.12 shows the S+NE paths for the parse tree and named entities that are presented in Figure 5.11.

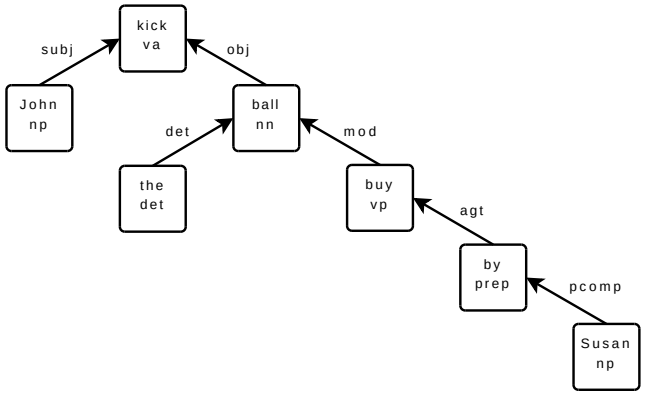
2. For every QPLM triple, we look for links between the $\mathcal{W}$ and $\mathcal{A}$ elements in all S+NE paths. If a match is found we copy the portion of the S+NE that contains the elements, change the element to their generic names ($\mathcal{W}$ and $\mathcal{A}$), and create a PRule for the QPLM conversion. For instance, as shown in Figure 5.13 every QPLM triple of Figure 5.11 can be found in the S+NE paths of Figure 5.12; the paths between all $\mathcal{W}$ and $\mathcal{A}$ are recorded as a PRule as shown in Figure 5.14.

3. Every PRule contains a frequency count that is computed over every sentence of the training corpus (PropBank mapped to QPLM). For instance, if the sentence "*Maria eats*" is analysed after the example of Figure 5.14, the frequency count of PRule (a) will be incremented by one. This is because they have similar S+NE paths and QPLM triples:

Original:
John kicked the ball bought by Susan.

QPLM:
$\langle Who(kick) \rightarrow\ John\rangle$, $\langle What(kick) \rightarrow\ ball\rangle$,
$\langle Which(ball) \rightarrow\ buy\rangle$, $\langle Who(buy) \rightarrow\ Susan\rangle$

Syntactic Dependency Trees:



Named Entities:
<ENAMEX Type=NAME> John </ENAMEX> kicked the ball bought by <ENAMEX Type=NAME> Susan </ENAMEX>.

Figure 5.11: The information types used for training are QPLM triples, syntactic dependency trees and named entities. We demonstrate these types using the original sentence "*John kicked the ball bought by Susan*".

> they describe a subject relation between a person and an active verb, creating a QPLM triple of the type $\langle Who(\mathcal{W}) \rightarrow\ \mathcal{A}\rangle$.

After computing all the training files, we would have a resulting PRule file containing all possible S+NE paths that can generate the manually-defined triples. If an S+NE path cannot be found then a PRule cannot be generated and the current training triple is skipped. Because the learning process basically consists of matching paths and counting frequencies, the training process is very fast and is equivalent to the time it takes to read the input files. In the forementioned Intel(R) Pentium(R) 4 machine, the training time is slightly over 2 minutes.

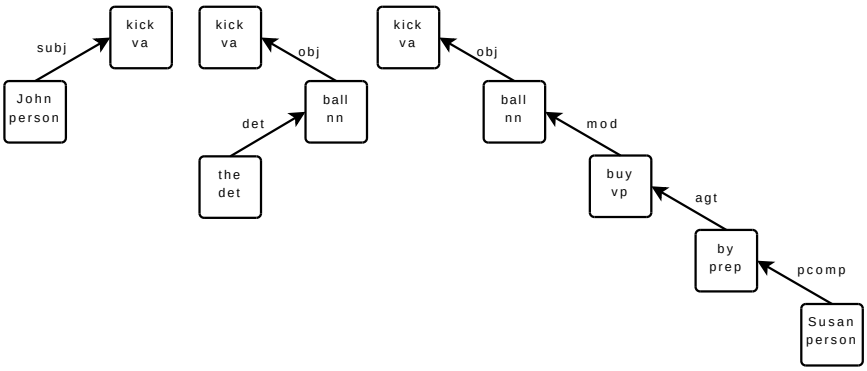


Figure 5.12: S+NE Paths created from the syntactic tree and named entities of Figure 5.11

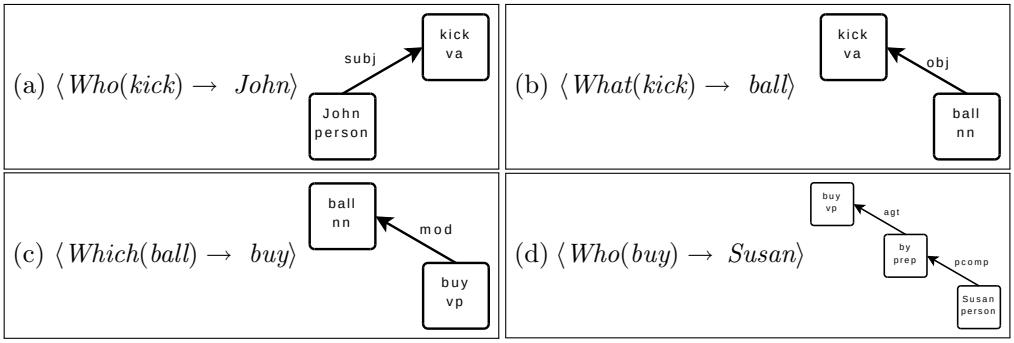


Figure 5.13: Matches of QPLM triples in S+NE Paths

5.3.3 Applying QPLM

Using the training corpus described above, we found all the PRules needed in order to generate the semantic triples when having an S+NE representation. Figure 5.15 illustrates this process. The rules are grouped by QPLM triples, having their S+NE paths attached with a frequency value. This frequency value represents how many times an S+NE path is used to create a PRule in the training corpus.

To convert S+NE files into QPLM, we start by applying the most important PRules, which are those with the highest frequency values. It is also important to observe that if an S+NE path generates different QPLM triples, we only need to apply the one with the higher frequency. For instance, if

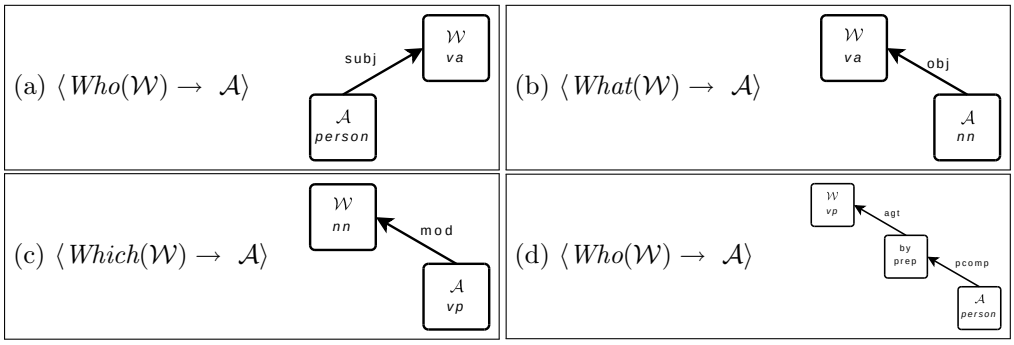


Figure 5.14: PRules created from the matches of Figure 5.13

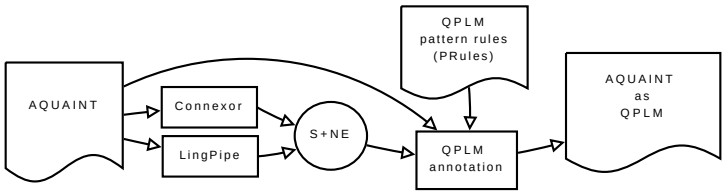


Figure 5.15: QPLM annotation

the pattern ${w_{person}}^{subj} \Rightarrow a_{va}$ is associated with the triple $\langle Who(\mathcal{W}) \rightarrow \mathcal{A} \rangle$ with frequency of 8 and with the triple $\langle Where(\mathcal{W}) \rightarrow \mathcal{A} \rangle$ with a frequency of 2, the S+NE path will only generate the *Who* triple. Because frequency is the decisive factor, we have 20% chance of wrongly assigning an incorrect semantic label.

We observed that more precise PRules could be created taking into account that some verbs generate a different QPLM triple for the same S+NE path. These new PRules (which we refer to as FW) are defined with a fixed $\mathcal{W}$ becoming less frequent but at the same time more precise. The precision of FW rules combined with the generality of the previous ones (which we refer to as GN) assure us that we have a correct analysis of a known verb as well as fair guess of an unseen one. To ensure that known verbs are evaluated first by the more precise FW rules, we assign a much higher weight to those rules than GN rules. An evaluation using the combination of both types of rules has shown that assigning a weight 800 times higher to FW than to GN gives us the best results.

We have noticed that many PRules will not be activated during annotation because they are shadowed by more frequent (and thus more important) rules. We remove these rules, which accounted for 12% of the total number of PRules. We have also observed that a large number of PRules have a frequency of one, which in many cases represent either noisy or very specific information. We remove these rules because they negatively impact the speed of the system and do not contribute much to the accuracy. Inevitably, the lower number of PRules causes a decrease in recall, which is more salient when taking into account the FW rules. Even though we experience a decrease in recall, removing low frequent PRules causes the removal of abnormal PRules that are generated by parsing errors. This causes the PRules set to be reduced to 56% of its original size.

Using this training set, we performed a 10-fold cross-validation. We trained using one part of the corpus and tested using the other nine parts; we repeated the process until all the ten parts are used for training. This has showed us that our annotation has a recall of 24.6%, where recall is the percentage of QPLM triples in the training set that are correctly generated by the annotation process.

The reduction of the training set to improve the speed, combined with the fact that 68% of the PropBank mapping could not generate rules due to broken parse trees, resulted in an upper limit of recall of 38%. This means that the obtained recall of nearly 25% is not far from the hypothetical upper limit, which suggests that the best way to improve the QPLM annotation is through optimizing the training set.

The presented recall values are measured using existing relations in the training set. We define precision for QPLM as the percentage of triples that are correctly annotated. Therefore, precision in this setting can only be measured manually since the annotation technique can generate QPLM triples that are correct but are not present in the training set. On average, QPLM generates three times the amount of triples of its PropBank representation. This includes relations that are not present in the PropBank representation

Triple	**Freq. × PRule**	**Example Sentence**
$\langle Where(\mathcal{W}) \rightarrow \mathcal{A} \rangle$	$288\times$ $\boxed{\substack{\mathcal{A} \\ eh}} \xrightarrow{\text{cla}} \boxed{\substack{\mathcal{W} \\ va}}$	*Under*$^{\mathcal{A}}$ *stress, some people cannot work*$^{\mathcal{W}}$ *efficiently*
$\langle HowMuch(\mathcal{W}) \rightarrow \mathcal{A} \rangle$	$105\times$ $\boxed{\substack{\mathcal{A} \\ nh}} \xrightarrow{\text{pcomp}} \boxed{\substack{to \\ eh}} \xrightarrow{\text{goa}} \boxed{\substack{\mathcal{W} \\ va}}$	*John drives*$^{\mathcal{W}}$ *to the beach*$^{\mathcal{A}}$

Figure 5.16: Examples of badly selected rules that cause a decrease in performance

such as relations between components of a noun phrase and including verbs that are not the main verb of the predicate. A hand analysis of 40 annotated documents has shown that 50.3% of the triples are correctly assigned.

During this evaluation, we have found that many errors made by QPLM are due to badly selected rules. We postulate that manually correcting or removing these rules can largely improve the precision of QPLM. Figure 5.16 shows some of these rules. For instance, we found a rule, which existed 288 times in the training corpus, that specifies that the initial position of an adverbial clause (*cla*) creates the triple $\langle Where(\mathcal{W}) \rightarrow \mathcal{A} \rangle$. Another rule defines that if we find a prepositional complement (*pcomp*) from the word "to" when it specifies a goal (*goa*) from the main verb, the relation should be defined as $\langle HowMuch(\mathcal{W}) \rightarrow \mathcal{A} \rangle$. These are clearly not good rules since they create QPLM triples such as $\langle Where(work) \rightarrow under \rangle$, and $\langle HowMuch(drive) \rightarrow beach \rangle$.

We realise that the performance of QPLM, in terms of recall and precision, can be greatly improved. Future improvements may include the use of different syntactic parsers, named-entity recognisers, as well as the modification of the set of semantic relations. Nevertheless, the relations found by QPLM when used with IR have shown to improve its results, in particular when applied to the QA task, as we will see in Chapter 6.

5.4 Concluding Remarks

This chapter presented the definition of an annotation model based on semantic relations between sentence words. We have compared our rule based

model, QPLM, against semantic role models such as FrameNet, PropBank and the semantic relation triples of Litkowski [77]. We have demonstrated how PropBank is mapped to QPLM triples so that conversion rules are obtained for our system. We have discussed how QPLM is used, its performance and its accuracy.

We have experienced difficulties due to errors of the Connexor parser. As future work, we could use a different parser or integrate different parsers as in the study of Nowson and Dale [104] to maximise the useful part of the training set.

QPLM has room for improvement, in particular in regards to precision and recall. We obtained a precision of around 50%, which appears to follow current measurements of precision for SRL systems when considering unseen predicates. However, this manual evaluation is performed over one of the test sets of the 10-fold cross validation, and we are unsure of the amount of unseen predicates in each of the validation sets. Future work will be carried out in regards to this evaluation.

Despite the fixed question format of the QPLM annotation model, there are possibilities to introduce extensions to its semantic model, where different questions or labels can be added or removed. It is likely that different models will provide different results, however it is important to stress that QPLM was kept simple so that it could generate semantic triples much faster than existing models. Although the addition of more complex labels or resolution algorithms could add extra representability to QPLM, the overall advantage might not justify the drop in speed performance. The evaluation of our work is presented in Chapter 6 including a comparison between the usage of QPLM and SRL for IR and QA.

6

Evaluation

QPLM represents a sentence as a collection of triples $\langle \mathcal{Q}(\mathcal{W}) \rightarrow \mathcal{A} \rangle$, where each triple specifies a question-style semantic relation between pairs of sentence terms. This chapter evaluates whether relations of this type help to find better answers for questions when used in the IR stage of a QA system. The evaluation is done through comparison with standard IR techniques and using them within QA systems.

Even though NLP has helped IR to achieve results that are more precise over standard methods [139], NLP has failed to convince IR researchers of its practical application. The main reason for this is that IR systems are intended to be used as online processes, with speed performance being one of the main issues; however, NLP research seems to focus more on accuracy than on speed. The different speed/accuracy standpoints of IR and NLP researchers separate both research areas. This chapter focuses on answering our second research question of whether linguistic information such as syntactic dependencies, semantic role labels and QPLM can provide better indication of relevance than BoW in IR for QA. This chapter is organised as follows: Section 6.1 describes the setup of the experiments used to evaluate the usage of QPLM in IR. Section 6.2 presents the results and Section 6.3 presents an analysis of the results along with future work.

6.1 Experimental Setup

QPLM generates triples containing relations between sentence words. In Chapter 5, we hypothesise that these relations assist IR in finding documents that contain better answers for natural language questions; we will test this hypothesis by experimentation. We have performed several experiments to test the impact of QPLM on IR and on QA, and how it compares to other similar methods such as syntactic relations and PropBank-style semantic roles.

6.1.1 Corpus and Question Set

The basis of our evaluation is taken from the 2004, 2005 and 2006 QA Tracks of TREC [32, 161, 162], which are standard testbeds for QA system comparison. The resources made available from these tracks provide a way of testing systems outside the conference competition; of these resources, we use the provided questions and the AQUAINT corpus. The AQUAINT corpus is the collection of documents containing 375 million words from three different sources of newswire data: The New York Times News Service (NYT), The Associated Press Worldstream News Service (APW), and The Xinhua News Service (XIE). All experiments presented in this chapter make use of the AQUAINT corpus. Unless otherwise stated, the research presented in this chapter is performed on the full extent of the AQUAINT corpus. The exceptions explicitly indicate the document set used and the reasons for using it.

Besides the AQUAINT corpus, we also use the evaluation scripts created by Ken Litkowski, which were made available at the TREC website[1]. These evaluation scripts can be used for either *lenient* or *strict* evaluations. In the *lenient* evaluation, a document is labelled as correct if it contains the answer pattern. On the other hand, the *strict* evaluation requires an answer to be retrieved from a document that is known to contain a correct answer. As

[1] `http://trec.nist.gov/data/qamain.html`

previously reported in the literature [75], and as we will describe shortly, the evaluation scripts that are provided after the conference do not give the same correctness of results that the participation in the conference does. However, because of the lack of other automatic means of evaluating QA results of TREC, we focus on these evaluation scripts and propose some modifications to it, so that the results are more reliable.

6.1.1.1 Issues with the Evaluation Patterns

The evaluation scripts are built from the pool of answers obtained from the responses of all participant QA systems in the respective years. The method of automatically evaluating systems through pooling the results of TREC competition has been regarded as effective in the IR field [158, 177]. The pooling method uses the relevance judgements made from the top-n results of each system that participated in the competition to create a list of relevant documents. This list can be used to evaluate systems that have not participated in the competition. However, Lin [75] describes such techniques as ineffective for QA, citing reasons such as the use of unsupported answer patterns and shallow pool depth of answers and supporting documents. Consequently, the comparisons resulting from this evaluation may not represent a precise grade of effectiveness of QA systems that did not participate in the original QA track.

Despite the problems with the pool depth of the answer patterns, QA is normally evaluated using the strict and the lenient measures. However, for the task of IR for QA, strict metrics have a high likelihood of being unreliable because of three main reasons. Firstly, the supporting document list for factoid questions was built based on one single answer and one supporting document for every system. Because many of these systems use a pre-compiled set of documents per topic given by NIST, as shown by Monz [97], there may be a bias towards the first documents of this list, and any modification to the ranking order of the top documents in this retrieval list will likely to be evaluated as bad, even though it might in fact improve the QA process.

Secondly, the pool of supporting documents cannot indicate the relevance of a document that has not been manually evaluated, making it impossible to measure the effectiveness of any different document retrieval technique which aims at finding new and unseen documents. Thirdly, the pool of supporting documents is small,[2] which causes precision and coverage values to be quite small and differences among techniques to be perceived as not to be statistically significant. Despite these issues with the strict evaluation for IR techniques, in addition to the standard lenient results, we will also present the strict evaluation for some of our results so that both measures can be more easily used in comparison with other research.

Other problems can be found with the TREC scripts for the lenient evaluation for QA. They do not evaluate whether an answer from a unknown document correctly provides the justification for an answer, nor do they consider different ways that the answer can be phrased. Therefore in some instances there can be errors where the evaluation script may consider a correct answer as incorrect and vice-versa. For instance, the answer pattern *Sarah Polley* for question 215.4 "*Which actress appeared in two films shown at the festival?*" regarding the topic "*1999 Sundance Film Festival*" is found in nearly 70 AQUAINT documents but less than five of those documents are related to the question. Because participant systems have only selected the most common usage of the answer string, the answer pattern *Kurt Cobain* for question 11.1 "*Who is the lead singer/musician in Nirvana?*" will fail to define other documents as relevant if they contain different spellings of the answer such as *Kurt Donald Cobain*.

We have found discrepancies between the answer patterns and the judgement file from TREC. For instance, question 81.5 has not been answered by any of the participant systems, therefore the judgement file does not have a valid known answer string, nor a supporting document. However, an answer pattern for this question is made available in the evaluation scripts. Further

[2]Considering only factoid questions that have been found to contain at least one supporting document, the average number of supporting documents per question is 3.7 for the combined questions of 2004, 2005 and 2006.

investigation shows that the answer pattern is the correct answer and that it can be found in the AQUAINT article NYT19990515.0255. This finding may be a manual attempt to provide the correct patterns for every question, even when the answer was unknown from the pool of answers.[3]

Another discrepancy found between the patterns and the AQUAINT files refers to the normalisation that systems perform on numbers and dates. Because some QA systems normalise numeric data, the output of the system might not match the content of the files. Since the answer patterns are based on this output, some answers might be judged incorrectly if the system being evaluated does not perform the same type of normalisation found in the judgement files. This can be seen clearly with question 187.3 "*How long is the Amazon River?*" which is evaluated by the regular expression *(6,500 kilometers|4000 MILES)*. There are three documents that support the answer, with two of those containing the phrase *4,000 miles* (with a comma separator) whilst there is no document in the collection that refer to the answer as *4000 miles* (without a comma separator).

6.1.1.2 Towards a Better Evaluation

The manual verification of answers is the most reliable evaluation possible; however, the cost involved is prohibitive, in particular when the evaluation needs to be repeated for different systems and setups over large corpora. Thus in our research where we are faced with this scenario, we perform the analysis of results automatically in spite of the known problems in such evaluations. In order to minimise these problems we presented our results using the whole set of 1,488 questions from TREC's QA track of 2004, 2005 and 2006, and a reduced set of questions whose answers are less likely to occur in a randomly selected document.

[3]According to one of the thesis reviewers, the answers are compiled from the pool of answers and from the answers found during the development of the questions. Ken Litkowski makes a specific reference in the answer patterns documentation of TREC 2006 that answers not found by any system were also included, however he does not discuss his methodology for 2004 and 2005.

Because the answer patterns originated from answers supported by the documents from AQUAINT, we can precisely evaluate the probability that an answer pattern occurred by chance in the retrieval set. For instance, if a known answer appears in only in one document in the entire AQUAINT, there is a 100% chance that finding the answer string in this collection represents finding a document that supports the answer. Therefore, the use of this question and answer pattern in this collection is 100% reliable. On the other hand, the presence of an answer pattern that appears frequently in documents cannot be used reliably. One such case is the answer pattern *London*, which is used to define whether a document answers the question 167.1 "*In what city is the Millennium Wheel located?*" It is possible to calculate the probability of relevance of documents that contain a certain answer pattern if we know the total number of documents that contain this answer pattern. Equation (6.1) defines the probability of relevance of a document d to correctly answering a question q if it contains an answer pattern p.

$$Rel(d, q, p) = \frac{\#\text{ documents that contain pattern } p \text{ and answers } q}{\#\text{ documents that contain pattern } p} \quad (6.1)$$

However, it is a difficult task to find the exact number of documents that contain an answer pattern that correctly answers a question, in particular in a large set of questions and documents, and with very frequent answer patterns such as numbers, dates and common locations. Nevertheless, we can estimate the lowest probability by assuming that every answer pattern has a minimum of one document that correctly answers the question. The lowest probability of relevance $MinRel(d, q, p)$ is calculated as shown in Equation (6.2):

$$MinRel(d, q, p) = \frac{1}{\#\text{ documents that contain pattern } p} \quad (6.2)$$

In order to build this reduced set of questions, we used the answer patterns for the factoid question and verified how often they occur in the AQUAINT corpus. This gives us two advantages: we can create a set of questions that lead to a more reliable evaluation, and we can build an evaluation scheme where each question has its own level of confidence.

We have decided to approach the evaluation automatically using different subsets of question sets and normalised evaluation metrics. The question sets are defined according to the type of the question and the frequency of occurrence of their answer pattern. Table 6.1 presents the different question sets and their descriptions.

Question Set Name	Size	Description
FULL	1,448	All question from TREC's QA track of 2004, 2005 and 2006.
FACTOID	995	Subset of FULL containing only fact-based questions.
FACTOID-A	925	Subset of FACTOID where answer patterns are available for all its questions.
FACTOID-01%	593	Subset of FACTOID-A where the answer pattern for its questions occur in less than 1% of the AQUAINT corpus. This means that there is less than 1% chance of a random selected file be evaluated as correct for a certain question.
FACTOID-TRUSTED	60	Manually created subset from FACTOID-01% where the answer pattern is known to generate trustable answers. We have defined an answer pattern as trustable where $P(a_j\|q_j)$ is at least 90%. This set of questions is listed in Appendix B.
FACTOID-1A	28	Subset of FACTOID-TRUSTED where the answer pattern can only be found in one AQUAINT document. Some questions that might have a single answer in AQUAINT were not included because the answer could not be found by an exhaustive regular expression search in AQUAINT. These questions are likely to fit into the aforementioned regular expression discrepancies. The questions in FACTOID-1A are indicated in the FACTOID-TRUSTED list in Appendix B.

Table 6.1: Question sets used in the evaluation

The questions from FACTOID-TRUSTED were manually selected from

FACTOID-A after performing a verification of the presence of the answer pattern in every document of the AQUAINT collection. This process was used to generate FACTOID-01% and FACTOID-1A, using the frequency of the occurrence of the answer patterns. The selection of FACTOID-TRUSTED was performed by manually inspecting questions and answers in the documents where the answer pattern is found. We included a question in FACTOID-TRUSTED if its answer pattern correctly answers the question for all documents in which it appears. For instance, for question 189.6 (TREC 2005) regarding the author *Joanne Kathleen Rowling* asks, "*In what city was Rowling born?*". The correct answer pattern *Chipping Sodbury* can be found in two articles in the AQUAINT corpus (NYT19980813.0007 and NYT20000112.0203), but only the latter document directly relates to *J. K. Rowling*, and provides the context to answer the question. Questions like these have been excluded from FACTOID-TRUSTED.

Please note that the creation of these sets of questions focuses exclusively on the problem of the low reliability of the answer patterns to indicate a correct answer. The usage of these sets can increase the confidence in the evaluation of an IR system, however there is no correlation between how rare a certain answer pattern is in a corpus and the difficulty of finding it. It is possible that frequent answer patterns represent questions that are harder to find and therefore questions that are more challenging to the IR system, however this correlation is not clear. Although we believe that these new sets of questions represent a step forward to fix some of the problems with the automatic evaluation for QA, a better evaluation set would not only contain the reliability of its answers but also the difficulty of its questions.

Since these questions are not always self-contained and in some cases do not constitute a proper natural language sentence (e.g. the question type *Other*), we performed some modification to the phrasing of the question so that they can stand independently without ambiguity. These modifications include the substitution of key pronouns with the topic, as well as the inclusion of the whole topic text when the topic is referenced in a briefer way.

Topic:	Gordon Gekko
Question:	What year was the movie released?
Modification:	Regarding Gordon Gekko, what year was the movie released?
Question:	What was Gekko's profession?
Modification:	What was Gordon Gekko's profession?
Question:	Other
Modification:	Tell me more about Gordon Gekko.

Figure 6.1: Examples of modifications applied to TREC questions

In some more difficult cases, when no substitution is possible and the question does not mention the topic, we add a phrase containing the topic at the start of the question. Some examples are presented in Figure 6.1; the complete set of self-contained questions is listed in Appendix B. Please note that even with the applied modifications, the questions of type *Other* are not self-contained questions. The question modification allow these questions to be used in our IR system, but they are still hard to evaluate. The evaluation presented in this chapter does not use such questions.

6.1.2 Evaluation Metrics and Statistical Significance

We evaluated the different IR methods using standard IR and QA metrics. Since QA is a task that uses a fixed number of documents, it is pointless to analyse the performance of a question set using all documents returned by an IR system. Therefore, all the metrics are used as though the retrieval set constitutes only a certain number of documents. For instance, if a QA system only uses 10 documents, we measure the performance of the top-10 ranked documents returned by the IR tool. In our experiments all questions returned at least n documents, which causes precision to be equal to redundancy normalised by n. Hence, the results shown in this chapter will be mostly presented in precision at rank n ($p@n$) and coverage at rank n ($c@n$). Definition of this metrics were previously defined in Chapter 2 on page 52.

We have performed statistical significance tests over the obtained results so that we can determine whether the differences in the results are due to chance. We performed paired tests using the answers found by each question as independent observations. In order to compare the results of QA systems, we paired the results for each question and assigned a score of 1 for correct answers and 0 for incorrect ones.

A comparative study by Smucker et al. [137] on statistical significance tests for differentiating IR results indicated that some tests should be avoided (Wilcoxon signed-rank test) while some (student t-test) give acceptable results even though they assume a distribution not found in the IR result. We follow the recommendation of Smucker et al. to use paired permutation tests to compare IR results.

A paired permutation test calculates the probability that the difference between two groups of paired results A and B is due to chance (null hypothesis). In order to reject the null hypothesis, the difference in means of A and B will occur with a lower frequency than a threshold, which is normally set to 5%, considering all possible pairwise permutations of the results for A and B. To compute the statistical significance it is necessary to derive new groups of results A' and B'. All possible A's and B's are created by either choosing the result for a certain question from A or from B. After the creation of these systems the difference in means between A' and B' is calculated and recorded. The exact probability for accepting or rejecting the null hypothesis is the p-value, which is the percentage of the recorded means that have a mean difference greater than the original mean difference.

The computational power required to calculate all permutations of a set of results is extremely large, especially for a set of questions as large as the one we have available to us. The number of possible permutations for an n pairwise test is 2^n, therefore the number of possible permutations for a set of 995 questions is 2^{995}, which is approximately 10^{300}. Since it is not practical to obtain an exact probability for this large number of permutations, we calculate an approximation by generating a large number of

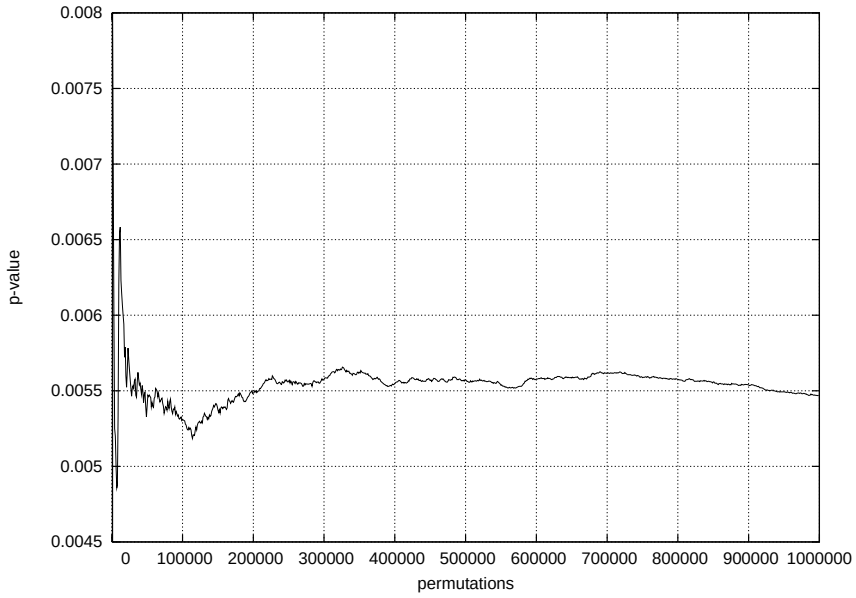


Figure 6.2: Variance of the p-value according to the number of permutations

random permutations. For our statistical significance tests, the number of random permutations is chosen at 100,000. This number of permutations is enough to give a precise p-value to the level of defining a 95% and 99% significance threshold. Figure 6.2 shows the variation of the p-value for a paired random permutation test up to one million permutations. The two retrieval sets tested are different in terms of statistical significance with a p-value measured as 0.005468 in the last permutation. As observed in Figure 6.2, the precision of the p-value is sufficient to establish that the samples are not significant on either 95% and 99% thresholds before the 100,000 permutation mark. The p-value at the 100,000 permuation was measured as 0.00531 differing from the p-value at the one million permuation mark by merely 0.000158.

In this chapter, we use the following symbols to represent the statistical significance of the results: we use the symbol $\blacktriangle$ when a significant improvement with $p < 0.01$ occurs, and the symbol $\triangle$ for $p < 0.05$; and when the results are significantly worse, we use the symbol $\blacktriangledown$ for $p < 0.01$ and $\triangledown$ for $p < 0.05$. We also indicate with the symbol $\circ$ when the difference in results

Syntactic Relations	contains	*BoW*
SRL	uses	*Syntactic Relations*
	contains	*BoW*
QPLM	trained with	*SRL*
	uses	*Syntactic Relations*
	contains	*BoW*

Figure 6.3: Relationship between techniques used

are not statistically relevant on a minimum of 5% level (i.e. $p > 0.05$).

6.1.3 Information Retrieval Framework

Chapter 4 described the framework of the IR tools used in our experiments. We applied the described methods to implement different indices and retrieval strategies in order to evaluate QPLM against related work such as BoW, Syntactic Relations, and SRL.

Because we are working with different types of relations (QPLM, Syntactic Relations, and SRL), we expect different sets of results for the retrieval sets; at the same time the most variance in results is normally obtained via different lexical features, which should be the same for all techniques that we have implemented. Therefore, two large retrieval sets obtained from these different techniques are likely to contain the same documents, but since we evaluate the top retrieved documents, it is possible to observe differences in the results. These differences are due to the different textual relations used; analysing these differences and finding if and how much is the improvement over standard IR giving by the use of QPLM is the goal of this research.

The interdependencies among the techniques (as shown in Figure 6.3) are likely to affect the degree of statistical significance of their results. In fact, the underlying dependencies influence the effectiveness of the technique, for instance, the failure of the syntactic parsing would negatively affect the subsequent SRL and QPLM. The actual extent to which the quality of dependent processes influences the results can be studied as future work.

We implemented distinct indices for each piece of relational information

that we used. The BoW index does not include any special relations beside the implicit co-occurrence; however, it does include the position of the words, which could be used to rank documents according to word proximity. We decided to keep all techniques as similar to each other as possible by using simple but comparable methods, so that differences in the retrieval sets are caused by the differences in the relation and not by the ranking algorithm.

QPLM and Syntactic Relations are used in the same way because they use similar annotation. Both techniques represent a direct relation between two words in a sentence. The QPLM and Syntactic Relation indices include information of the specific relation number in a document, the relation type and which role the word plays in the relation. The roles for the syntactic relation are either head or modifier, while in QPLM they are $\mathcal{W}$ or $\mathcal{A}$ as part of the triple $\langle \mathcal{Q}(\mathcal{W}) \rightarrow \mathcal{A} \rangle$.

The representation of the PropBank type semantic roles requires the indexing of relations not only between words, but also between whole phrases. In this way, the existence of the relation will be represented in all words of the related phrases. Every word informs which relation in the document it is part of, the role in the relation, and the position among the other words in the related phrase. Even though we do not make use of the positional information of words in the phrase, we include the information in case a proximity ranking is to be implemented.

For all of the different indices that we implemented, if a non-stopword does not belong to any relation, the word is included as part of an empty relation. This means that it will be included in the index, but the word is only related to the document. This allows a term to be found and to be used even if it is not part of any relation. This feature ensures that all types of indices will contain the same amount of lexical items as the standard BoW.

6.1.4 Question Answering Systems

To better understand the relation between the sets of retrieved documents and QA, we applied the sets on the following four QA systems.

- *Aranea*: Developed by Lin [76], the Aranea system utilises the redundancy from the World Wide Web using different WWW search engines. The system relies on the text snippets to generate candidate answers. It applies filtering techniques based on intuitive rules, as well as the expected answer classes with named-entity recognition defined by regular expressions and a fixed list for some special cases.

- *OpenEphyra*: Developed by Schlaefer et al. [130], the OpenEphyra framework aims to be a test bench for QA techniques. The system uses a standard approach for QA: using a three-stage QA architecture (Question Analysis, Information Retrieval, Answer Extraction), it employs WWW search engines on its IR stage and maps the answers back to the TREC corpus. The system is among the top-10 groups and has an accuracy of 20.8% for the factoid component at the main QA Track at TREC 2007 [30].

- *MetaQA*: We developed the MetaQA [111] system, which, similarly to the Aranea QA system, makes heavy use of redundancy and the information provided by WWW search engines. However, it goes a step further by combining different classes of WWW search engines (including WWW Question Answering Systems) and assigning different confidence scores to each of the classes.

- *AnswerFinder (AF)*: Developed by Mollá and Van Zaanen [96], the unique feature of AF is the use of QA graph rules learned automatically from a small training corpus. These graph rules are based on the maximum common subgraph between the deep syntactic representation of a question and a candidate answer sentence. The graphs are derived from the output of the Connexor dependency-based parser [143].

For most of these systems, some modifications of the standard system configuration were required. All the systems studied, with the exception of AF, make heavy use of WWW search engines and the redundancy obtained

Size	BoW	QPLM
128 chars.	4.80	6.12
256 chars.	*8.73*	*10.12*
512 chars.	7.56	9.21
Full Doc.	4.06	5.21

Table 6.2: MRR scores obtained by the Aranea QA system for different snippet sizes when used with the BoW and QPLM on the NYT set of documents

to find their answers. For our experiments, we had to turn the WWW search off, causing a significant decrease in performance when comparing to the reported results in the literature. Because AF's IR component is performed offline, the integration is seamless and only required to provide the system with a list of documents in the same format as TREC. The OpenEphyra framework is neatly designed and implemented, however the interaction between its components still depended on the overall system architecture, which made the implementation of new modules for the system quite difficult.

With the exception of AF, all the QA systems received a retrieval set as a collection of snippets. This is because these systems are based on WWW Retrieval and they expect to receive documents in this format. For every document, we extract a fixed size window of text where more question words (non-stopwords) are found. An evaluation of this passage retrieval method only using the NYT portion of AQUAINT is shown in Table 6.2. This evaluation has shown us that the 256-character window with the current snippet construction method gives the best results when used on the Aranea QA system.

6.1.5 Overall Experiment Framework

Figure 6.4 illustrates the connection between the IR system and its different setups and QA systems, and the different evaluations that were carried out. Figure 6.4 shows three main types of processes:

- **Indexing**: where the text from the corpus is processed by one or more text processing modules according to the type of index that is going

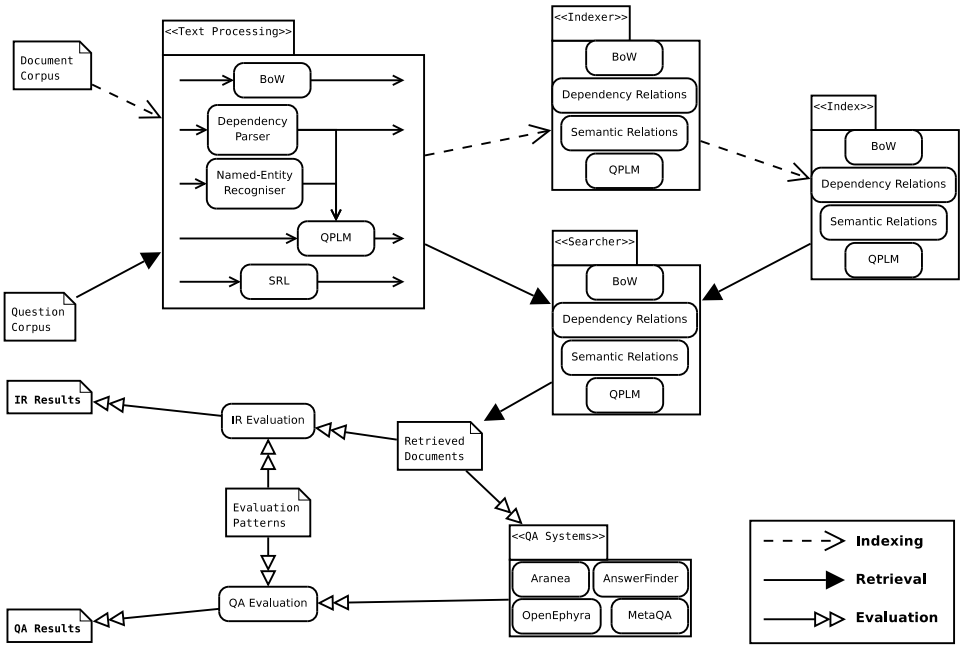


Figure 6.4: Framework of the experiments

to be built.

- **Retrieval**: where the questions are processed in the same way as the corpus was processed during indexing and by using the proper index, a set of documents is retrieved.

- **Evaluation**: where the retrieved documents are processed by two different types of evaluation: IR evaluation, which solely involves the retrieved set of documents; and QA evaluation, which involves a set of QA systems. Both use the same set of evaluation patterns.

6.2 Results

This section details the experimental results obtained by applying QPLM in IR for QA. We first focus on the retrieval set generated by the IR using QPLM only. The evaluation is performed intrinsically and extrinsically; the intrinsic evaluation focuses on the primary function, which is to answer a question; and the extrinsic evaluation focuses on the function or setup

n	$p@n$		$c@n$	
	lenient	(strict)	lenient	(strict)
1	41.51%	(11.32%)	41.51%	(11.32%)
5	36.69%	(7.61%)	65.62%	(26.49%)
10	32.85%	(5.69%)	73.41%	(35.36%)
50	24.68%	(2.34%)	87.35%	(54.91%)
100	21.75%	(1.49%)	90.27%	(63.89%)

Table 6.3: Lenient and strict evaluation of retrieval set for FACTOID-A using the AQUAINT corpus with the QPLM approach

purpose, which for a retrieval set is to find documents that contain an answer to a question [63].

Later in this section, we shift our attention to the comparative results of QPLM with the standard BoW retrieval approach, and to linguistically-oriented methods using syntactic dependency relations and semantic role labels.

6.2.1 Information Retrieval Results

Table 6.3 presents the results of the retrieval set using FACTOID-A with the QPLM approach. The evaluation is considered *lenient* when a document is defined as relevant if it contains the required answer string. On the other hand, the *strict* evaluation defines that only previously known documents that contain answers are considered relevant.

The first row (n=1) in Table 6.3 shows that QPLM is able to answer 384 out of 925 questions (41.51%) from FACTOID-A using only the first retrieved document based on the presence of an answer string. This means that QPLM when used on IR would satisfy the requirements of a QA system for more than 40% percent of questions if no further processing to find an answer were required. However, only 11.32% of this documents are previously known to contain the answer. Considering that many QA systems rely on several documents to provide an answer, it is possible to observe in the lenient evaluation that 90.27% of the questions can be answered when using 100

n	FACTOID-01%		FACTOID-TRUSTED	
	p@n	***c@n***	***p@n***	***c@n***
1	29.85%	29.85%	15.00%	15.00%
5	26.27%	56.16%	14.67%	35.00%
10	22.98%	66.27%	12.50%	45.00%
50	15.11%	84.65%	05.30%	60.00%
100	12.20%	88.87%	03.73%	68.33%

Table 6.4: Evaluation of retrieval set for FACTOID-01% and FACTOID-TRUSTED using the AQUAINT corpus

documents, and 21.75% contain a correct answer. On the other hand, for the strict evaluation the precision at 100 documents is extremely low since the list of documents containing answers to questions is small.

It is important to note that although the lenient evaluation might represent an overestimation of results due to its tendency to select more documents than only those that deal with the question at hand, there might be some documents with the correct answer that were defined as incorrect because the answer pattern is different. Therefore, the lenient result does not mean an upper bound in performance, however the strict evaluation considers only those few documents that have been previously manually analysed, and in this way it certainly defines a lower bound in performance. As previously discussed in Section 6.1.1, the lenient method of evaluating retrieval sets automatically may produce many false positives and false negatives. In order to minimise this problem we use FACTOID-01% and FACTOID-TRUSTED, which are reduced and more reliable sets of questions. Table 6.4 present the results using these question sets.

As expected, the coverage and precision values shown in Table 6.4 are lower than those presented in Table 6.3. Although FACTOID-01% and FACTOID-TRUSTED contain fewer questions than FACTOID-A, the contribution of each question/answer is more significant. The precision measure seems to be largely affected by the usage of this set of questions. This is explained by the smaller frequency of occurrence of the answer patterns

associated with the question set. A smaller frequency means that fewer documents are relevant; therefore, the chance of finding a relevant document is smaller, as well as for some questions the list of relevant documents will be quickly exhausted. This exhaustion can be observed because that the difference in coverage is not as salient as the difference in precision. For instance, we observed similar c@10 for both FACTOID-A and FACTOID-01%.

The size of the retrieval set has a strong impact on the QA results. A small number of documents means that fewer questions have the possibility of being answered (lower coverage and redundancy) at the same time that those documents are more likely to contain the correct answer (higher precision); while, a large amount of documents means that more questions have the possibility of being answered at the same time that those documents are less likely to contain the correct answer. Ideally, the IR system would have a perfect precision returning just the right amount of documents required by the QA system to generate the correct answer. However, to date IR systems are far from generating flawless results.

QA systems can generate wrong answers even when using perfect IR results. This occurs due to a number of reasons, including grammatical errors leading to an erroneous sentence interpretation, questions types that are hard to extract, and even answers that are phrased in an atypical way that makes their extraction difficult. A more comprehensive evaluation strategy would account for query difficulty and document set complexity; until then, we obtain an overall assessment of the QPLM performance for QA by comparing the results of different QA systems.

6.2.2 Question Answering Results

The results obtained by the QA systems were built using the resources available at the TREC website, such as the regular expressions with the factoid answers and the evaluation patterns made by Ken Litkowski. The score of the results described in this study present represent the percentage of questions whose answers match the correct answer pattern. For all systems except

QA System	Factoid Accuracy
Aranea	8.06%
OpenEphyra	9.55%
AF	5.43%
MetaQA$^{c@10}$	1.51%

Table 6.5: Factoid results for c@1 on QA systems using FACTOID-A

the MetaQA system, the results are presented as c@1; for the MetaQA system, we present the results as c@10 (coverage at answer 10) since MetaQA is not designed to return one single answer. MetaQA is a WWW-based system that work as an aggregator of information sources and its ranking mechanisms only work when sufficient evidence is given for certain entities. MetaQA returns pages of answers and hence we evaluate the system based on its first page, which consists of 10 results.

Table 6.5 shows the results obtained using QPLM's retrieval set on different QA systems. We observed that OpenEphyra is the best performing system, answering nearly 10% of the questions. This reinforces our previous statement that the results of the QA systems are not to be directly compared to those reported in the literature. A smaller snippet size, even though they might influence positively in the QA results, as shown in Table 6.2, are less likely to contain the correct answer. Therefore, if the FACTOID-A question set measured a precision of 24.68% at rank 50, it is a good finding that a QA system provided a correct answer for nearly 10% of questions when provided with fifty snippets (one per document).

6.2.3 Model Comparison

In this thesis, we hypothesised that linguistic information provides a better indication of relevance for documents in regards to answering a natural language question. This section draws a comparison between the retrieval methods using different term dimensions, such as QPLM, BoW, syntactic dependency relations, and semantic role labels. In this set of comparisons, we made sure to maintain equivalent lexical items for all the methods so that

	FACTOID-A		FACTOID-01%		FACTOID-TRUSTED	
	***p*@10**	***p*@50**	***p*@10**	***p*@50**	***p*@10**	***p*@50**
BoW	26.96%	21.10%	18.13%	12.54%	10.67%	04.77%
QPLM	*32.85%* ▲	*24.68%* ▲	*22.98%* ▲	*15.11%* ▲	*12.50%* ▲	*05.30%* ▲
diff.	+21.85%	+16.96%	+26.79%	+20.49%	+17.19%	+11.19%

Table 6.6: Precision for BoW and QPLM using the AQUAINT documents

the variance found in the results is solely attributable to the differences in the models.

6.2.3.1 Comparison with Bag-of-Words

Most off-the-shelf IR systems implement a retrieval strategy based on the occurrence of keywords in the document. This strategy is called Bag-of-Words (BoW), since all words that are not stopwords are equally chosen to represent the document in the index. Most current IR ranking strategies extend the BoW approach by adding the position of the word to try to maximise the density of query words in a window of text; however, these density-based metrics are normally applied as a re-ranking strategy of the standard BoW approach. In our approach, QPLM and BoW have the same lexical items, meaning that the difference observed in the results of both methods is only due to the extra information supplied by the QPLM relations. Table 6.6 compares the precision obtained for the BoW and QPLM retrieval using the TREC question set and the AQUAINT documents.

We can observe from the results presented in Table 6.6 that QPLM performs consistently better than BoW. The measured improvement is as large as 26.79% for FACTOID-01%, 21.85% for FACTOID-A, and 17.19% for FACTOID-TRUSTED. The mean difference is statistically significant at the 1% level for all results. This comparative study shows that BoW provides us with less relevant documents than QPLM, however the difference between QPLM and BoW seems to decrease when more documents are used. This can be visualised in Figure 6.5. For a large enough retrieval set, the

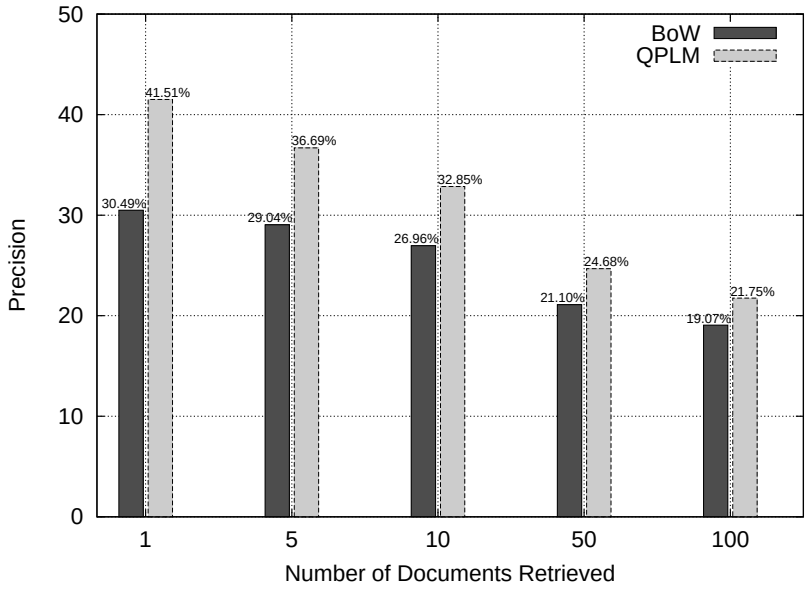


Figure 6.5: BoW and QPLM results for FACTOID-A

differences between BoW and QPLM will become zero, since both methods use the same lexical features, and the differences are due to the different ranking produced by the systems.

QPLM also dominates over the BoW rankings when taking into account coverage values, as it can be observed in Table 6.7. In a similar way to the precision metric, the largest improvements are in high ranked documents and on the FACTOID-01% and FACTOID-TRUSTED question sets. Coverage and precision in the FACTOID-01% set is somewhat lower than FACTOID-A, which may not be a negative finding, since FACTOID-A results may account for a significant larger evaluation noise than the other sets due to overly frequent evaluation/answer patterns; problems that FACTOID-01% and FACTOID-TRUSTED address. In this sense, the results for FACTOID-01% and FACTOID-TRUSTED show that the QPLM improvement is consistent on the different subsets of the TREC question set.

Table 6.8 shows the strict evaluation for QPLM and BoW. It can be observed that the differences in the results are very small and not statistically significant between QPLM and BoW. This is expected as the number of

	FACTOID-A		FACTOID-01%		FACTOID-TRUSTED	
	***c*@10**	***c*@50**	***c*@10**	***c*@50**	***c*@10**	***c*@50**
BoW	63.57%	80.11%	54.81%	74.70%	40.00%	51.67%
QPLM	*73.41%*	*87.35%*	*66.27%*	*84.65%*	*45.00%*	*60.00%*
	▲	▲	▲	▲	△	▲
diff.	+15.48%	+9.04%	+20.92%	+13.32%	+12.50%	+16.13%

Table 6.7: Coverage for BoW and QPLM using the AQUAINT documents

	MRR	***p*@10**	***p*@50**	***c*@10**	***c*@50**
BoW	0.1852	5.64%	2.25%	*35.40%*	53.15%
QPLM	*0.1931*	*5.69%*	*2.34%*	35.36%	*54.91%*
	◦	◦	◦	◦	◦
diff.	+4.28%	+1.02%	+3.86%	-0.11%	+3.31%

Table 6.8: Strict evaluation for BoW and QPLM using the AQUAINT documents

known documents is small.

We also evaluated the impact of these retrieval sets intrinsically by applying them to different QA systems. The measurements used to extrinsically evaluate retrieval sets for QA do not strictly match the actual usage of the retrieval set. Table 6.9 shows the differences of coverage in the results when the QPLM and BoW retrieval sets are used in QA. We can observe that despite the numerical improvements given by QPLM in the QA systems results, only few of them are statistically significant. Considering the combined set of questions, we obtained significant improvements when the QPLM retrieval set was used with the OpenEphyra and the MetaQA systems QPLM. This shows that the QPLM model can help the QA task.

The set of questions in FACTOID-1A is much more restricted. It constitutes only 28 questions whose answer patterns are found only once in the entire AQUAINT corpus. For this restricted set of questions, as shown in Table 6.10, we can observe that even though QPLM does not significantly improve the coverage, it creates a document ranking where the document that correctly answers a question appears earlier in the list. The low cover-

QA System		2004	2005	2006	Total
Aranea	**BoW**	7.83%	7.46%	7.69%	7.74%
	QPLM	*8.26%*	*7.46%*	*8.28%*	*8.06%*
		◦	◦	◦	◦
	diff.	+5.56%	+0.00%	+7.69%	+4.17%
OpenEphyra	**BoW**	6.96%	8.56%	7.94%	7.94%
	QPLM	*8.70%*	*10.77%*	*8.93%*	*9.55%*
		◦	△	◦	▲
	diff.	+25.00%	+25.81%	+12.50%	20.25%
AF	**BoW**	3.91%	6.08%	4.47%	4.92%
	QPLM	*4.35%*	*6.91%*	*4.71%*	*5.43%*
		◦	◦	◦	◦
	diff.	+11.11%	+13.64%	+5.56%	+10.20%
MetaQA (*c*@10)	**BoW**	0.43%	1.10%	0.50%	0.70%
	QPLM	*0.87%*	*2.76%*	*0.74%*	*1.51%*
		◦	▲	◦	▲
	diff.	+100.00%	+150.00%	+50.00%	+114.29%

Table 6.9: Comparison of results in *c*@1 for BoW and QPLM using QA systems on FACTOID-A

age of both techniques are related to the use of common words in the topic and the question, as well as the lack of stemming or other lexical substitution/equivalency methods.

6.2.3.2 Comparison with Syntactic Relations

QPLM is a rule-based semantic annotation scheme built on top of a syntactic representation that includes named entities. It is important to understand whether similar improvements experienced by QPLM over BoW can be achieved by using only syntactic relations. We postulate that syntactic relations have the capacity to improve the results of an IR system and in particular the results of QA systems. The inclusion of syntax in IR allows a better ranking for documents that hold similar syntax to a question. We assume that this will preserve noun phrases and other important syntactic constructions, allowing better documents to be retrieved.

	c@100	*MRR*	Average Position
BoW	42.86%	0.036245	27.59
QPLM	46.43%	*0.048202*	*20.75*
	◦	△	▽
diff.	+8.33%	+32.99%	-24.81%

Table 6.10: Evaluation of retrieval set for FACTOID-1A using the AQUAINT corpus

The lenient results presented in Table 6.11 highlight the differences in results for BoW, syntactic dependency relations and QPLM. We obtained the dependency relations with the Connexor syntactic parser [143]. We ran out of computer memory when running the results for the combined document set, and as the problem also occured with the NYT corpus, we have decided to present the results only on the APW and on the XIE portions of AQUAINT separately. The evaluation of the FACTOID-A set of questions has shown that on the APW corpus, QPLM improves the IR results on both precision and coverage. The improvement over syntactic relation is statistically significant. Syntactic relations can be used to improve the results of BoW; however, they only have shown to improve on the coverage metric, while for precision the results are not statistically significant at a 5% level. For the XIE corpus, the differences between the results are not statistically significant for any of these techniques. The differences obtained for the strict evaluation are also not statistically significant.

The differences observed in Table 6.11 are only partially observed when the retrieval sets are applied to QA systems. The QA results shown in Table 6.12 indicate that there is a significant improvement when using Aranea over syntactic information in the APW documents. The numerical results of the QA systems are higher when using syntactic information; however, the difference is statistically significant only when using OpenEphyra in APW. Similarly with the intrinsic evaluation of Table 6.11, the results are not significantly different for the XIE part of AQUAINT.

		p@100 lenient	*p*@100 (strict)	*c*@100 lenient	*c*@100 (strict)
APW	**BoW**	13.61%	(0.29%)	68.65%	(19.35%)
	Syntax	13.74%	(0.52%)	70.92%	(29.45%)
		◦	◦	△	◦
	diff. BoW	+0.92%	(+80.97%)	+3.31%	(+52.18%)
	QPLM	*15.39%*	(0.58%)	*73.73%*	(33.04%)
		▲	◦	△	◦
	diff. Syntax	+12.01%	(+11.84%)	+3.96%	(+12.19%)
XIE	**BoW**	*9.20%*	(0.35%)	52.65%	(22.92%)
	Syntax	9.02%	(0.36%)	*53.19%*	(18.27%)
		◦	◦	◦	◦
	diff. BoW	-1.92%	(+3.36%)	+1.03%	(-20.29%)
	QPLM	8.98%	(0.47%)	52.97%	(23.55%)
		◦	◦	◦	◦
	diff. Syntax	-0.44%	(+27.44%)	-0.41%	(+28.87%)

Table 6.11: Precision and coverage for syntactic relations on FACTOID-A

6.2.3.3 Comparison with Semantic Role Labelling

We used the SRL tool SwiRL from [142]. This tool has good precision and coverage, however, as shown in Chapter 5 on page 121, it is slow and quite unstable when parsing large amounts of data. We have assembled a cluster of computers in order to speed up the corpus annotation, but even with many dedicated computers the estimated completion time is longer than one year. The lack of SRL systems available to quickly annotate a large amount of data motivated the use of a simplified and quicker technique. We used the QPLM annotation tool which takes less than 3 weeks to fully annotate the 3 GB of data from the AQUAINT corpus using a single machine.

Since we wanted to determine how QPLM compares to SRL, particularly on the basis of its usage for IR and for QA, we performed some tests using the available amount of data annotated with semantic roles. The part of the AQUAINT corpus annotated includes the first 41,116 documents, in chronological order, from the New York Times (NYT) newspaper.

QA System		APW	XIE	QA System		APW	XIE
Aranea	**BoW**	5.38%	4.52%	AF	**BoW**	3.19%	2.35%
	Syntax	5.81%	5.38%		**Syntax**	3.94%	2.65%
		◦	◦			◦	◦
	diff. BoW	+8.00%	+19.05%		**diff. BoW**	+23.53%	+12.86%
	QPLM	7.74%	4.52%		**QPLM**	*4.36%*	*2.73%*
		▲	◦			◦	◦
	diff. Syntax	+33.33%	-16.00%		**diff. Syntax**	+10.50%	+2.89%
OpenEphyra	**BoW**	5.73%	5.33%	MetaQA (c@10)	**BoW**	0.60%	1.45%
	Syntax	7.14%	6.23%		**Syntax**	0.83%	1.58%
		△	◦			◦	◦
	diff. BoW	+24.56%	+16.98%		**diff. BoW**	+38.16%	+9.37%
	QPLM	8.14%	5.33%		**QPLM**	1.18%	1.40%
		◦	◦			◦	◦
	diff. Syntax	+14.08%	-14.52%		**diff. Syntax**	+42.34%	-11.53%

Table 6.12: Comparison of results in c@1 for BoW, syntactic relations and QPLM using QA systems on FACTOID-A

Using these documents and the questions from TREC, we retrieved a set of 50 documents for every question. We analysed the impact of the semantic annotation when used on document indices by checking the presence of the answer string in the documents returned. We also obtained a list of 50 documents using solely the BoW approach in order to compare the gain over standard retrieval.

Table 6.13 presents the results of the retrieval set for FACTOID-A using the BoW, the SRL and the QPLM approaches. As we can observe, the SRL approach gives the best results for both p@50 and c@50. Even though the numeric results for c@50 are equivalent, the retrieval set and the questions that both techniques are able to answer are different. For instance if the FACTOID-A is broken into years, the coverage metric for 2004 and 2005 favour SRL, while the coverage for 2006 favours QPLM. The difference of the results for SRL and QPLM on both precision and coverage are not statistically significant, while the difference found between SRL and BoW is highly distinct. The results from the strict evaluation are not statistically significant.

	***p*@50**		***c*@50**	
	lenient	(strict)	lenient	(strict)
BoW	12.44%	(0.36%)	57.41%	(13.91%)
SRL	*14.08%*	(0.40%)	*60.76%*	(14.47%)
	▲	◦	▲	◦
diff. BoW	+13.23%	(+11.04%)	+5.84%	(4.01%)
QPLM	13.27%	(0.37%)	*60.76%*	(13.52%)
	◦	◦	◦	◦
diff. SRL	-5.76%	(-9.04%)	+0.00%	(-6.56%)

Table 6.13: Comparison of BoW, SRL, and QPLM for FACTOID-A

	***p*@10**	***p*@50**	***c*@10**	***c*@50**
BoW	24.17%	19.40%	64.42%	89.54%
SRL	*28.04%*	*21.97%*	*73.69%*	94.77%
	▲	▲	▲	△
diff. BoW	+16.05%	+13.23%	+14.40%	+5.84%
QPLM	26.44%	20.70%	69.65%	*94.77%*
	◦	▼	▼	◦
diff. SRL	-5.71%	-5.76%	-5.49%	+0.00%

Table 6.14: Comparison of BoW, SRL, and QPLM for FACTOID-01%

We also performed an evaluation on the FACTOID-01% question set. The results, presented in Table 6.14, show the same trend of performance as Table 6.13 where SRL outperformed BoW and QPLM. The FACTOID-01% results do show a significant difference between QPLM and BoW at p@50. On the other hand, in Section 6.2.3.1 we have shown this difference with a more comprehensive test involving the complete AQUAINT corpus. It can also be observed in Table 6.14 that there is no significant difference between the results for QPLM and SRL in terms of c@50.

In a similar way to previously described, we evaluated the retrieval sets according their usefulness to QA systems. The results presented in Table 6.15 show that analogous to the results obtained for the retrieval sets alone, the numerical results of the QA results are higher for SRL than QPLM and BoW. The differences of the results are significant in the comparison between SRL

QA System		c@1	QA System		c@1
Aranea	**BoW**	2.91%	AF	**BoW**	1.61%
	SRL	3.52% ◦		**SRL**	2.41% ▲
	diff. BoW	+20.69%		**diff. BoW**	+50.00%
	QPLM	3.62% ◦		**QPLM**	2.21% ◦
	diff. SRL	+2.86%		**diff. SRL**	-8.33%
OpenEphyra	**BoW**	3.72%	MetaQA (c@10)	**BoW**	1.91%
	SRL	5.23% ▲		**SRL**	2.81% ◦
	diff. BoW	+40.54%		**diff. BoW**	47.37%
	QPLM	4.52% ◦		**QPLM**	1.81% ▽
	diff. SRL	-13.46%		**diff. SRL**	-35.71%

Table 6.15: Comparison of results in c@1 for BoW, SRL and QPLM using QA systems on FACTOID-A

and BoW for the OpenEphyra and the AF systems, and between SRL and QPLM in the MetaQA system. Nevertheless, the difference between QPLM and SRL is not significant for all QA systems evaluated using c@1. Overall, QPLM performs slightly worse than SRL but still, on average, outperforms the BoW technique.

The experiments with the QA systems show that SRL performs slightly better than QPLM; a statistically significant difference was only observed between SRL and QPLM for the MetaQA system. We observed a clear improvement when using SRL to BoW in both AF and OpenEphyra. It is important to point out that results for the retrieval set alone in Table 6.16 showed BoW outperforming QPLM on redundancy for the 2004 questions. This could be indicative that OpenEphyra answer-extraction modules are more precise than the other QA systems and do not heavily rely on redundancy as do the Aranea and the MetaQA systems; however, redundancy seems to be the factor that influenced the QA results for the 2006 questions, since QPLM achieved a higher coverage for that year.

Year	Method	Factoid Score	*c*@50	*r*@50
2004	**BoW**	2.17%	33.33%	2.92
	SRL	*3.04%*	*35.33%*	*3.20*
	QPLM	2.17%	34.47%	2.79
2005	**BoW**	5.52%	41.13%	5.02
	SRL	*7.46%*	*43.77%*	*5.50*
	QPLM	6.35%	42.08%	5.29
2006	**BoW**	2.98%	34.57%	3.65
	SRL	*4.47%*	36.33%	*4.37*
	QPLM	4.22%	*38.45%*	4.16

Table 6.16: Correlation between redundancy and factoid scores for the OpenEphyra QA system

Because of the high dependency on WWW sources, the MetaQA system performed poorly. As explained earlier, due to design differences the results are measured using c@10 instead of c@1. Nevertheless, even with the non-conformity of the system, it appears to support our viewpoint that semantic annotation can enhance the IR results for QA. Not surprisingly, since the MetaQA is a redundancy-based QA system we observed that its QA results follow the redundancy results more closely than the coverage ones.

Results also show that AF correctly answered only a few questions for the given question set. On the other hand, it provided some consistent results such that the improvements are due to additional correct answers and not to a larger but different set of correct answers. AF showed a similar performance for both semantic-based strategies and both outperformed the BoW strategy.

6.3 Concluding Remarks

This chapter described the setup of the experiments and presented the evaluation of QPLM in comparison to related work. We observed from these experiments that QPLM improves the relevance of IR for QA without presenting an excessive cost in regards to speed performance. Therefore, we

have demonstrated that QPLM, and possibly other similar methods of semantic representation that are fast enough to satisfy the IR requirements, can be effectively used in IR to improve QA.

The comparison between QPLM and BoW showed that QPLM outperforms this standard technique; however, QPLM still performs worse than SRL. SRL is a more complex and complete type of representation. The association of phrases in SRL create better retrieval relations than the association of words in QPLM. However, the unsatisfactory speed performance renders SRL unusable for IR. The trade-off between speed and quality of results favours QPLM over SRL since QPLM's large advantage in speed compensates the small difference of IR and QA results.

The results presented in this work for all QA systems are much lower than those reported in the literature. This undesirable but expected problem occurred not only because of the modifications that we had to perform in the systems to make them appropriate for our experiments, but mainly because of the reduced number of documents used for this evaluation. Another reason for the difference in the results, in particularly of the OpenEphyra QA system, is that the results reported in the literature are possible because of the use of some proprietary components which were removed in the system that was made public.

Only recently we have been able to test Koomen et al. [71] SRL tool. This SRL tool is the top ranking SRL tool at the CoNLL-2005 Shared Task Evaluation and it seems to be much faster than SwiRL. Preliminary tests suggest that it is able to perform the annotation of AQUAINT in almost one full year using a single computer; however, this tool, like SwiRL, is not very stable, crashing several times during our experiments. Further work would involve employing several computers and attempting to parse the whole AQUAINT corpus with this tool.

It is important to point out that although the tool of Koomen et al. seems much faster than SwiRL, QPLM still outperforms both of them on speed by a large margin. QPLM represents word relations that are built using rules from

syntactic and NE information. This simpler representation, combined with a smaller number of supporting NLP tools, allow QPLM to be faster than current SRL tools. Further work on the QPLM tool could focus on increasing its performance on both speed and accuracy. QPLM's precision and recall figures can be improved by using a hand-annotated corpus. QPLM's speed suggests that it can be currently used on IR tools as a pre-processing engine. It is understandable that any delay in the IR phases is undesirable when dealing with large amount of data, therefore optimizing the speed of QPLM is one of our priorities.

We have shown that syntactic relations can be used in the IR stages of a QA system to improve the accuracy of the retrieval set. However, when using a QA system we only obtained weak improvements. As future work one could measure the effect of syntactic parsing errors in QPLM. For this, one could apply different syntactic parsers and NE recognisers on some gold standard QA set.

The lack of a gold standard retrieval set for QA makes the process of evaluating new QA systems and related techniques quite challenging. As future work, one could use our study for creating the different retrieval sets (FACTOID-A, FACTOID-01%, FACTOID-1A, FACTOID-TRUSTED) in order to create a gold standard retrieval question set and evaluation scripts. Further work could focus on finding all supporting documents for a number of questions and answers and building a significant corpus that can be reliably used to evaluate fact-based QA systems.

In answer to our second research question, we can state that linguistic information does help to obtain better documents than words alone, and that improvements are sustained when simpler semantic techniques are used. Therefore, this thesis bridges the gap between IR and NLP by showing that it is possible to create NLP processes that can be incorporated into IR without largely interfering in the speed performance and improving the overall results.

7

Final Remarks

In this thesis, we have investigated whether NLP techniques that are commonly used in QA can be effectively integrated into the IR stage of QA systems. In particular, we have investigated the use of named entities, syntactic dependency relations, semantic roles, and a novel approach to semantic relations called the Question Prediction Language Model (QPLM). We have also hypothesised that it is possible to improve QA by opening the IR black box and modifying its structure to better suit the QA task. We have proposed an IR framework that easily allows the indexing and retrieval of different types of linguistic information. We have empirically shown that the addition of the information encoded in the proposed framework can help QA systems find the correct answers more easily.

In the introductory chapter we presented two main research questions, which were addressed in the subsequent chapters of this thesis. We now present the summarised conclusions to these research questions as follows.

1. *Can the presence of answer features such as named entities be used as an indication of relevance for documents in IR for QA?*

Chapter 3 investigated whether factoid questions that require certain types of named entities as their answer could use the presence of named entities of the same Expected Answer Type (EAT). We have developed a pseudo-relevance technique that extracts named entities of the EAT from a

retrieval set and creates a new retrieval set by assuming that this information is an indication of relevance. Even though it is clear that the absence of the named entity type required to answer a question represents the non-relevance of a document, we have been unable to show that the presence of such information indicates the relevance. We have shown that if the correct answer string is found in the retrieval set, the pseudo-relevance technique is successful in obtaining a better retrieval set. However, on average, the amount of noise introduced by unrelated named entities causes the feedback query to drift away from the original subject.

2. *Are syntactic dependencies and semantic role labels better indications of relevance of documents for QA than words alone? If so, can the same improvements be sustained with simpler semantic annotation?*

Chapter 4 proposes an IR framework that allows the addition of a wide range of linguistic information in the search data structure. We also describe a novel ranking mechanism that extends the standard vector-model similarity measurements to consider different dimensions that are built from the relational information. Chapter 5 proposes QPLM, which has shown to be a faster and simpler technique than current Semantic Role Labelling (SRL) tools. In Chapter 6, we have shown that the addition of linguistically-oriented dimensions to the standard bag-of-words dimension in a vector space retrieval allows results to improve in IR and in QA. We have also shown in Chapter 6 that QPLM sustains the improvements obtained by standard linguistic information such as syntactic dependency relations and semantic roles. QPLM provides improved results over dependency relations and even though QPLM has slightly worse numerical results in comparison to SRL, the difference is not statistically significant.

Therefore, we have shown that syntactic and semantic features can be integrated into the IR process and can serve as an indication of relevance for QA. We have also shown that improvements are sustained with simple semantic annotations such as QPLM. QPLM provides a better indication

of relevance for QA than syntactic information, and provides just as much indication as SRL, which is a type of semantic information that is more complex and more difficult to generate.

We have also shown that improvements in the retrieval set of documents are not always sustained with the same degree in the QA phase. This is another indication of the mismatch between the intrinsic and extrinsic evaluation of retrieval sets for QA, a point raised in Chapter 3 and in [112]. This supports the conclusion that better evaluation metrics and higher integration between IR and QA are needed.

In Chapter 4, we proposed a framework for the indexing and retrieval of relational information that is fast and uses small amounts of memory. We compared our framework performance with the representation using relational databases (MySQL) and an IR system built using the state-of-the-art Xapian IR toolkit. The comparison has shown that our framework is more suitable for the encoding of relational information such as syntactic dependency relations. We have obtained smaller inverted file structures and faster speeds on both the indexing and the retrieval stages.

7.1 Future Work

The IR framework proposed in Chapter 4 has shown to outperform a relational database and a state-of-the-art retrieval tool, in terms of indexing and retrieval speed, and in terms of disk space needed for its index structure. However, we believe that further improvements in this framework are still achievable. For instance, inverted file compression techniques have yet to be fully explored, and even though the framework allows distributed IR, we have not performed any experiment in this regard.

We believe that the IR framework is an important outcome of this thesis, and that we should contribute to the community by making it freely available. Some immediate work involves allowing the framework and some IR tools built over this framework to be available as an open source project.

Although we have shown that QPLM and SRL help to improve IR for QA

when comparing to the standard use of BoW, we have not yet implemented different ranking algorithms in our IR framework, nor implemented a passage retrieval option. Another possible promising study would be to apply our ranking technique or to use QPLM in the state-of-the-art IR tools.

SRL is seen as a promising auxiliary process for QA, however the improvement obtained by its usage is still small [135, 131] or even negative [8]. Although we have not presented large improvements with the usage of SRL, we have shown that similar improvements can be sustained with annotation tools that are simpler and faster. Therefore, we believe that the direction for research regarding semantic roles and IR for QA is not towards heavier and more comprehensive semantics but towards lighter and perhaps more restrictive methods.

QPLM has shown to equally help IR for QA, even though its accuracy is still lower in comparison to SRL tool. This suggests that the quality of the semantic annotation may not play the largest part in guiding IR systems into retrieving the best documents for QA. However, we believe that QPLM can prove to be even more useful if its precision and recall are improved. Further work towards improving QPLM includes using syntactic parsers and different training corpora.

We noticed that many errors originated from syntactic parser errors; the usage of different syntactic parsers or a combination of them, as employed by Nowson and Dale [104], may offset the errors in annotation and improve the overall results. Another interesting question, which deserves further investigation, is whether we can obtain the same annotation without the use of off-the-shelf syntactic parsers, and even obtain the syntactic rules that form the semantic annotation directly from the training corpus. By removing the dependency on off-the-shelf syntactic parsers, there is potential to further improve the annotation speed performance and its accuracy.

Even though QPLM labels resemble natural language questions, we believe that they do not need to be restricted to this representation. Further experiments could be carried out with the use of different types of seman-

tic labels, which can facilitate the use of available resources (such as the PropBank) and the comparison with standard techniques.

We also believe that QPLM can be used to pinpoint exact answers, in a similar way that semantic triples are used to extract answers in the work of Litkowski [77], however further experiments are needed to support this claim.

7.2 Thesis Contributions

To summarise, we have made the following contributions with this thesis:

- We have found that named entities of the same type that is expected of an answer to a question cannot be used as an indication of relevance of documents in a pseudo-relevance feedback technique.
- We have found that semantic relations can be effectively used in the retrieval stages of a QA system, providing improvements over the bag-of-words baseline.
- We have developed an IR framework that allows a wide range of linguistic information to be represented, with efficient usage of memory and fast speed performance.
- We have developed a semantic relation model that is more suitable for IR than current Semantic Role Labelling standards because it is simpler and faster to generate, whilst delivering similar overall QA results.

A

Tupi Framework for Information Retrieval

The lack of support for NLP information from IR tools makes it hard for researchers to integrate these techniques. In this chapter, we propose the Tupi IR framework that allows the inclusion of different types of information such as those produced by NLP tools. This information is quickly stored and accessed since it can be included in the search structures of the IR systems built with the aid of the proposed framework. This chapter is divided into three sections: Section A.2, which details the Tupi framework for IR and the details and interaction between its components; and Section A.3, which highlights the construction of the IR system named Jemu. The Jemu IR system allows the inclusion of different types of relational information. Finally, in Section A.4 we present some concluding remarks and future directions for research.

A.1 Representing Relations in IR

IR has based its searching algorithms on the existence of words that capture the need of a user expressed by a set of keywords. A keyword search means that the token granularity of the search is at the word level. Different retrieval tasks may require different token granularity, for instance a word search tool based on specific sound descriptions could use similar keyword IR techniques but using phonemes instead of keywords. The choice or what

type of token is used is defined by the type of task and corpus.

If a more precise retrieval is needed, it becomes necessary to perform a further analysis in the documents/query. Such analysis could include linguistic information such as syntactic dependencies and semantic roles. The addition of extra linguistic information allow the construction of a different retrieval model than the standard BoW. For instance, the inclusion of syntactic dependencies distinguish between searching for "*comments about the candidates*" and "*comments about the election*". This models are only achievable if the linguistic information is added to the search index. Similar results could be achieved if the keywords used are of low frequency that returns a small amount of documents. However, with the growing amount of available information, low frequency words will still account for a large amount of document, which would demand a significant post processing effort.

A.2 Tupi Framework

The Tupi Framework for IR is designed to facilitate the construction of IR systems that incorporate relational information among the tokens of documents. Unlike previous IR research that incorporate this type of information, the Tupi Framework scales up well and is not largely affected by performance or memory issues. Tupi can represent different types of relational information whilst maintaining the simplicity of the BoW approach.

The Tupi framework is based on the concept that IR systems do not need to know how the data is internally organised, and like a database management system, they have the freedom to define what type of relational information will be included. Tupi is not intended to be a database system, nor does it allow the freedom of such systems, but it does focus on allowing IR systems to represent textual relations in a way that can be easily stored and retrieved.

The two main components in an IR system and in the Tupi framework are the *Indexer* and the *Retriever*. In our framework, these components together with other data structures need to be implemented in order to form

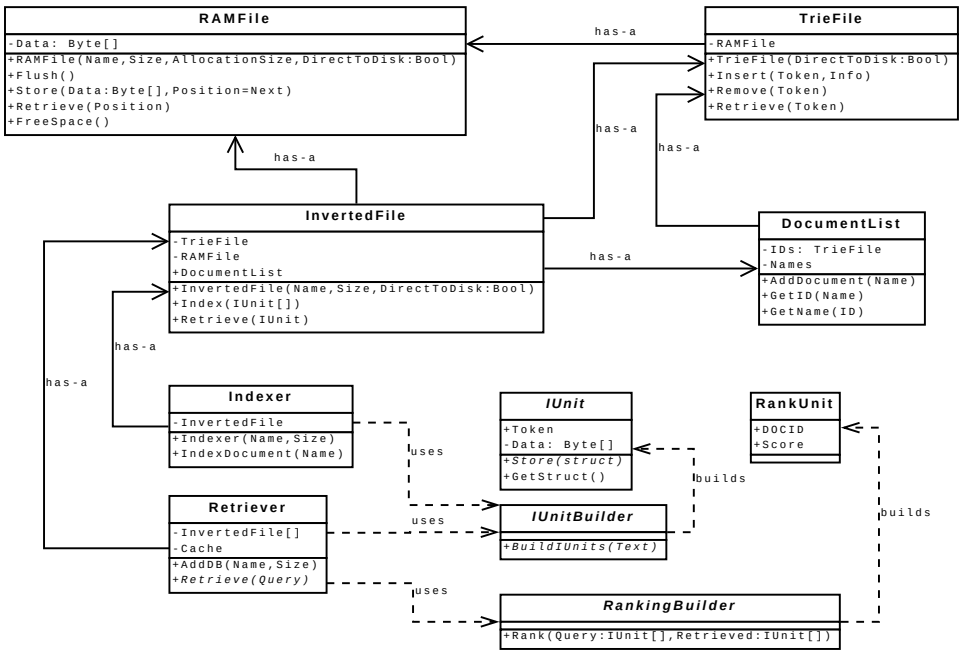


Figure A.1: Tupi framework

a complete IR system. Figure A.1 shows an overview of the framework and how the classes relate to each other. In this framework, both *Retriever* and *Indexer* classes use objects of the *InvertedFile* class. The *InvertedFile* class implements an inverted file, which stores the names of the indexed documents and their respective unique identification numbers via the class *DocumentList*.

The communication between *Indexer*, *Retriever* and *InvertedFile* is done via the abstract class *IUnit*. The translation between text to a list of *IUnits* is done by another abstract class named *IUnitBuilder*. The implementation of *IUnit* defines what type of information and how it is stored in the inverted file; the implementation of *IUnitBuilder* defines how the information is converted between text to *IUnit*. The coding of information from structured data to a sequence of bytes is done by the class *IUnit*, while the class *InvertedFile* is responsible for optimising the storage of *IUnits* for every indexed token.

The token position in the class *InvertedFile* is recorded using the class

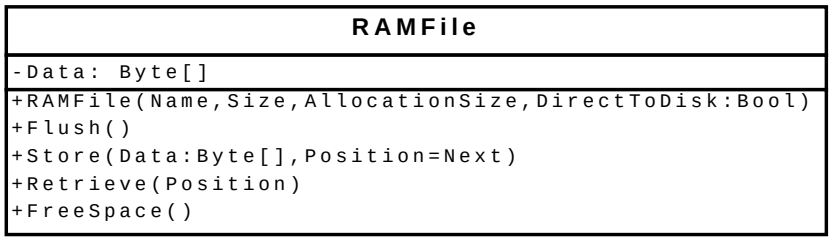


Figure A.2: *RAMFile* class

TrieFile. As the name of the class suggests, *TrieFile* implements a trie structure [27]. Both *TrieFile* and *InvertedFile* store information on disk via the class *RAMFile*. *RAMFile* allows the creation of files in RAM memory and only writes them to persistent storage when requested.

The modules of the Tupi Framework are built to allow more freedom of representation than the standard IR techniques, without compromising the performance of such systems. These modules are described in details in the following subsections.

A.2.1 *RAMFile*: The Interface with Memory and Disk

The *RAMFile* class, shown in Figure A.2, allows the communication of the other classes with persistent storage. Its constructor receives a file name, a file size and the data allocation size, and a flag indicating whether to bypass the memory buffer and read/write directly from the persistent storage. The allocation size is necessary since it defines where a position in the file is found. For instance, position 20 with an allocation size of 3 KB will start at 60 KB from the start of the file.

The DirectToDisk flag defines whether to bypass the memory buffer. It is used when the access to disk is not too frequent or when there is not enough main memory to support the whole file. Currently, *RAMFile* is only able to create files on RAM that are smaller than the available memory size; future developments will allow this file to be larger than the memory available by using paging techniques.

The main methods implemented by *RAMFile* are *Store*, *Retrieve* and

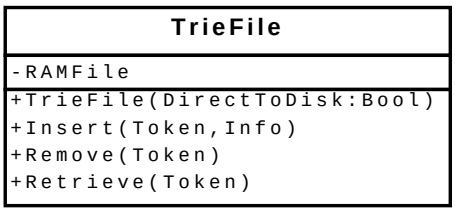


Figure A.3: *TrieFile* class

Flush. *Store* records a given data in the file either sequentially from the last recorded position or at a given position. *Retrieve* returns the data of a given position, and *Flush* transfers the RAM buffer into the persistent memory. Because of the pre-defined file size of the current implementation of *RAMFile*, *Retrieve* and *Store* will return an error if the sequential read/write reaches the end of file.

A.2.2 *TrieFile*: Structure for String Retrieval

The *TrieFile* class, shown in Figure A.3, implements a trie structure using *RAMFile* for the storage and retrieval of information. According to Clement et al. [27], a trie $T(S)$ is a data structure defined by a recursive rule $T(S) = \langle T(S/a_1), T(S/a_2), \ldots, T(S/a_r)\rangle$, where S is a set of strings over the alphabet $A = \{a_j\}_{j=1}^{r}$, and S/a_n are all strings of S starting with a_n and which are stripped of its initial letter.

In addition to the performance advantages of such structures, tries facilitate the use of stemming techniques, since the process of finding the set of words that start with a word stem is straightforward. Figure A.4 demonstrates the retrieving process of the words *connect*, *connected*, *connecting*, *connection*, and *connections* via the common stem *connect*, in a trie structure.

The *TrieFile* class contains the following methods: *Insert*, *Remove*, and *Retrieve*. The *Insert* method adds a token to the structure with associated information, *Remove* deletes a token and its information, and *Retrieve* returns the information associated with a particular token.

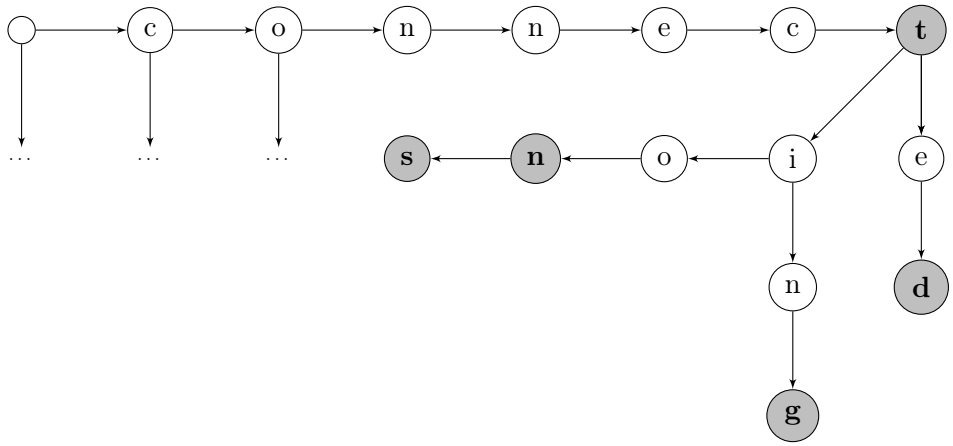


Figure A.4: Example of how a trie structure can be used to recover all words associated to word stem *connect*. Nodes in gray and bold characters represent word endings, where the address of the word in the inverted file is located.

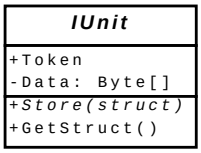
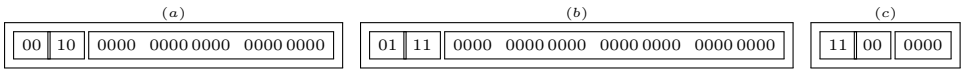


Figure A.5: *IUnit* class

A.2.3 *IUnit*: Defining the Data Structure

IUnit is one of the most important classes of the Tupi Framework and regards the data type and its organisation. *IUnit* is an abstract class, i.e. not implemented in the framework itself. The implementation of a class derived from *IUnit* defines the data structure that will be used in the IR system based on the Tupi Framework. Figure A.5 is the abstract class representation of an *IUnit*.

IUnit requires that the type of data structure implemented to be serialised into a byte stream and appended with a data preamble. The abstract class *IUnit* uses a sequence of bytes to store information and requires the implementation of two methods: *Store* that converts from a data structure



Type	Size	Bits		Data
00 (*a*)	10 (2)	20	(4 bits + 2 bytes)	0000 0000 0000 0000 0000
01 (*b*)	11 (3)	28	(4 bits + 3 bytes)	0000 0000 0000 0000 0000 0000 0000
11 (*c*)	00 (0)	4	(4 bits + 0 bytes)	0000

Figure A.6: Example of a data structure as defined by an implementation of the *IUnit* class

to a byte sequence, and *GetStruct* that converts the byte sequence into a data structure.

We have currently defined a 4-bit prefix that defines the type of information and the amount of bytes the information occupies. The first 2 bits define the type of information, while the other 2 bits define the size of the data in bytes (excluding the remainder bits of the first byte). This representation allows four distinct types of data to be represented. Figure A.6 shows an example of such representation.

The example in Figure A.6 shows that we have three types of data, we will call them *a*, *b* and *c*. Type *a* data starts with the prefix 00‖10 indicating the type on the first 2 bits (00) and the amount of trailing data bytes in the last two bits. For type *a*, we have 2 trailing data bytes, indicated by the bits 10. The data of *a* is the remainder bits of the first byte plus the amount of bytes indicated in the prefix, which accounts for 20 bits.

The data prefix indicating the size of the data is used to indicate where the next data package will start. This information is important not only for the data encoding/decoding stages, but also because it allows using less memory when there is no need for a larger chunk of memory. For instance, we need at least 19 bits to represents the document ID 262,144; therefore the representation of such information as type *a* data would be 00‖10 0100 0000 0000 0000 0000. On the other hand, we need only 4 bits to represent the document ID 14, which would render the following representation: 00‖00 1110. The size prefix can help compression techniques if needed.

According to this representation, the maximum amount of data bits is 28, which is enough for the purpose of this framework since it can represent 2^{28} distinct numbers. If all IDs are represented in one only inverted file with one term per document and no compression, the minimum amount of space needed to store this information is 1 GB ($2^{28} \times 4$ bytes). It is to be understood here that such representation is hypothetical and the number of terms per indexed file is certainly larger than the trivial example of one. In reality, if the number of IDs are exhausted the amount of memory needed will be many times larger than today's standards for any simple structure implementation. We understand that if one wants to assign a unique identification number for every document that is indexed, the 2^{28} limitation can be restrictive. Taking that the average document size in the GOV2 collection of documents used in the Terabyte Track [19] is nearly 18 KB, we can represent a data collection of 4.5 TB using 2^{28} distinct documents. This amount of documents is enough for today's existing fixed corpora, but is not sufficient for WWW indexing. Because of this, we designed the framework, so that unique IDs are assigned per inverted file. Therefore, we can represent an unlimited number of files by creating multiple inverted files.

A.2.4 *InvertedFile*: Efficiently Storing Information

One of the core structures that allow the development of an IR system is the inverted file. Since files are searched via keywords, the inverted file, which is a structure that indicates the location of all document keywords, is largely responsible for the performance of an IR system. What distinguishes the Tupi framework from other frameworks is that these keywords may contain more information than simply where they occur.

The class *InvertedFile*, shown in Figure A.7, represents an inverted file structure using a *RAMFile* for persistent data storage, a *TrieFile* for the token information, and a *DocumentList* for the document information. *DocumentList* is declared as a public object, so that the classes *Indexer* and *Retriever* can directly use its functionality and data. *InvertedFile* implements

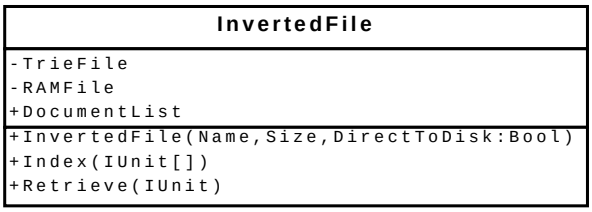


Figure A.7: *InvertedFile* class

two main methods: the *Index* method that is responsible for organising a number of *IUnits* in the inverted file structure, and the *Retrieve* method that deals with finding all *IUnits* associated with a given *IUnit*. This association can be defined as all *IUnits* of a given token, or as the *IUnits* that have the same token and share same other similar information.

The use of prefixes in the *IUnit* structure allows the representation of different data types in the same inverted file. Other IR strategies solve this problem by the use of multiple indices; a solution that would not require the prefix data, therefore reducing the size occupied by the inverted files. However, the use of multiple indices will likely require some redundant data structures, such as dictionaries.

Our aim with using this type of structure is not only to allow the representation of different types of information, but also to allow the representation of information that is dependent on each other. This dependency causes the records of the inverted file to be an inverted file on their own. For instance, we can mark the presence of a word in different documents of a corpus, in different sections of those documents, in different paragraphs of those sections, and so on. We define this dependent information as optional, meaning that if a word is part of a table and not a proper paragraph, it does not need to indicate which paragraph it pertains.

We represent different data types with different prefixes in order to provide some freedom of representation for different types of information in the inverted file. For instance, if we consider the previous types of information in Figure A.6 as document ID for a, related word for b, and word position for c, we can build an indexing scheme that every word contains the informa-

(a) *IUnit* representations as they appear on sentences

IUnit	**1**	**2**	**3**	**4**
Term(ID)	John(135)	Susan(145)	Susan(145)	Chris(187)
Document ID	25	25	25	25
Frequency	1	1	1	1
Related Word(ID)	Susan(145)		Chris(187)	
Relation(ID)	love(15)		love(15)	
Position	1	2	3	4

(b) Inverted file representation of (a)

Term(ID)	**Inverted File Data**
	(*a*-doc-freq) (*b*-relation-r.word) (*c*-position)
John(135)	(*a*-25-1) (*b*-15-145) (*c*-1)
Susan(145)	(*a*-25-2) (*c*-2) (*b*-15-187) (*c*-3)
Chris(187)	(*a*-25-1) (*c*-4)

Figure A.8: *IUnit* and inverted file representations for "*John loves Susan*" and "*Susan loves Chris*"

tion of which documents it appears, which words it relates to, and in which document position it is located. Because b contains 28 bits of data, it can only build the relation with 2^{28} words in the dictionary. Assuming that we do not work with more than 2^{20} words, we could use the first 8 bits of data to represent the type of relation. Therefore, the data of type b would inform the relation type and the related word. Type c can be used to represent the position of the word in the document. The existence of the word in a document can be recorded in many different ways; it can be a different type in the structure (as in c) or as part of a larger data structure, e.g. type a being a combination of document ID and Term Frequency. Either way, the type b data is clearly optional, which will allow the indexing of terms when they do not relate to other terms in the document. This representation, with type a being a combination of document ID and Term Frequency is exemplified in Figure A.8 for the hypothetical sentences/relations "*John loves Susan*" and "*Susan loves Chris*".

(a) *IUnits* for "*Susan likes John*"

IUnit	**1**	**2**
Term(ID)	Susan(145)	John(145)
Document ID	26	26
Frequency	1	1
Related Word(ID)	John(145)	
Relation(ID)	like(14)	
Position	1	2

(b) Inverted file representation of (a)

Term(ID)	**Inverted File Data**
	(*a*-doc-freq) (*b*-relation-r.word) (*c*-position)
John(135)	(*a*-25-1) (*b*-15-145) (*c*-1) (*a*-26-1) (*c*-2)
Susan(145)	(*a*-25-2) (*c*-3) (*b*-15-187) (*c*-4) (*a*-26-1) (*b*-14-145) (*c*-1)

Figure A.9: *IUnit* and inverted file representations when adding the relation/sentence "*Susan likes John*"

Figure A.8 shows the representation of the *love* relationship between John, Susan and Chris. As it can be observed, when no relationship is present, there is no need to include type b relations. Considering that the sentence and relationship "*Susan likes John*" appears in a following document, the *IUnit* and the inverted file representations will be as shown in Figure A.9.

What can be noticed in the example of Figure A.9 is that all data in the sequence of data refers to document ID 25 until document ID 26 is included. The same principle can be applied to other types of data if the different types are dependent on each other (e.g. document, paragraph, sentences and words).

Although we have not implemented compression techniques such as variable gaps, γ or σ codes [86, Chapter 5], the *InvertedFile* is able to incorporate them. Currently the only technique used to reduce the size of the inverted file is allowing data types to be represented with fewer bytes than the maximum when possible. The implementation of compression techniques is compatible

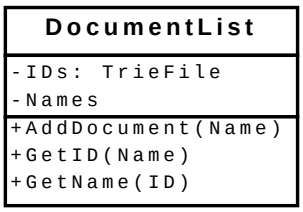


Figure A.10: *DocumentList* class

to the inverted file format and specifications.

A.2.5 *DocumentList*: Finding Indexed Documents

The *DocumentList* class, shown in Figure A.10, can be accessed as a public object of the *InvertedFile* class. We have defined it in this way to maintain the strong link between both structures. The *DocumentList* class represents a list of documents that have been indexed with a unique document ID. This class offers the following functionalities: *AddDocument* to include a new document, *GetID* to return the ID given a document name, and *GetName* to return the document name for a given document ID.

Currently *DocumentList* uses a sequential list of documents for returning the document name for a given ID, and a *TrieFile* for returning the ID for a given document name. The trie structure presents an efficient way to store documents with full directory path information, in particular when many files in the same directory need to be indexed and when they have a sequential naming as those in the AQUAINT corpus. The example of Figure A.11 shows the representation for the following AQUAINT files: NYT19990611.0023, NYT19990611.0025, NYT19980712.0080, and APW19990110.0072. As observed different documents are created for every news article, identified by a four-digit suffix.

The versatility of the trie structure and its linear performance is the main reason for its usage. However, when the file list is known, such as the one in the AQUAINT corpus, a better strategy would be to implement a hash function to convert between document names and document IDs.

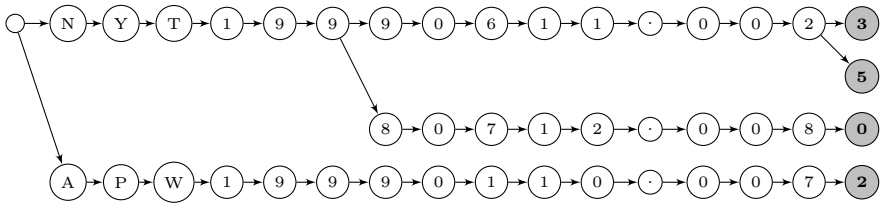


Figure A.11: Trie representation of the name of AQUAINT documents NYT19990611.0023, NYT19990611.0025, NYT19980712.0080, and APW19990110.0072

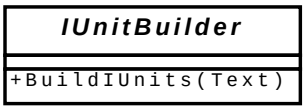


Figure A.12: *IUnitBuilder* class

The implementation of such functions can be easily done via overriding the implementation of the methods of class *DocumentList*.

A.2.6 *IUnitBuilder*: Extracting Meaning from Text

The abstract class *IUnitBuilder*, shown in Figure A.12, requires its only method to be implemented. An implementation of *IUnitBuilder* needs to convert textual information to a set of *IUnits*; therefore, this class is responsible for the necessary text processing steps of the Tupi Framework. For instance, if an IR system requires the implementation of a bigram model, the implementation *IUnitBuilder.Bigram* will be responsible for dividing the text into bigrams and creating the set of *IUnits* according to a pre-defined structure in the implementation of *IUnit.Bigram*. For more complex textual processing tasks, *IUnitBuilder* classes might require help from external NLP tools, such as syntactic parsers and named-entity recognisers. Figure A.13 is an example of some possible children implementations of the class *IUnitBuilder*.

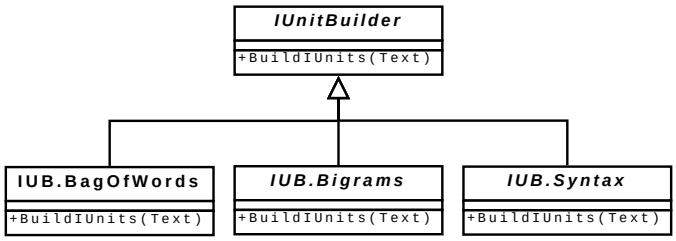


Figure A.13: Example of implementations of the *IUnitBuilder* class

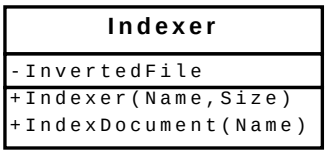


Figure A.14: *Indexer* class

A.2.7 *Indexer*: Processing and Organising Documents

The *Indexer* class, shown in Figure A.14, is responsible for reading documents and creating the inverted file via the *InvertedFile* class. The *Indexer* class receives a name to be assigned to the inverted file and its maximum size. These constructor parameters directly affect the classes *InvertedFile*, its *DocumentList* and consequently the *TrieFiles* and *RAMFiles* since this classes require this information in order to create their necessary storage files.

The *Indexer* class uses the method of *UnitBuilder* to create a set of IUnits that are going to be passed to the *InvertedFile* class for storage. The role of the *Indexer* implementation is to organise the data flow between these structures. The *Indexer* class works with only one inverted file at a time. After processing the document, the list of *IUnits* is passed to the *InvertedFile* object at hand, if the *InvertedFile* is unable to Index the information for lack of memory space, the *Indexer* class is responsible for closing the exhausted *InvertedFile* and creating a new one.

The *InvertedFile* class is not implemented as thread-safe, which means that if there is a need for faster indexing, several indexing processes can-

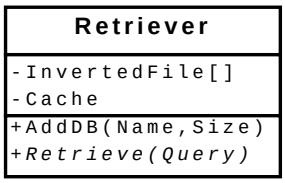


Figure A.15: *Retriever* class

not work with the same *InvertedFile* object. Nevertheless, it is possible for several indexing processes to be launched simultaneously in one or more machines creating different *InvertedFile* objects. One or more *InvertedFiles* can be used simultaneously by one or more *Retriever* processes.

A.2.8 *Retriever*: Finding the Best Documents

The main function of the *Retriever* class, shown in Figure A.15, is to utilise the structure of the *InvertedFile* to find documents that are relevant to a user query. The Retrieval class must interpret the query in the same way as the *Indexer* does for the documents, converting it to a set of *IUnits* via a specialised implementation of the class *IUnitBuilder*. The query *IUnits* are used to recover information from the *InvertedFile* that is similar to the ones in the query. This list of *IUnits* from the *InvertedFile* is then used along with the list of *IUnits* from the query to build a list of documents sorted by relevance. This ranking list of documents is built by an implementation of the abstract class *RankingBuilder*, which should return a list of Document IDs (DOCID) and their scores (*RankUnits*).

The *RankingBuilder*, represented in Figure A.16, is an abstract class that must implement a single function that combines query and the retrieved *IUnits* into a list of documents. This is normally one of the main components of a IR system; good ranking functions should push good documents to the beginning of the list and bad documents to the end of it. The Tupi framework allows the implementation of different ranking functions via the implementation of the *RankingBuilder*.

Because retrieval is not a sequential task like indexing, the process can

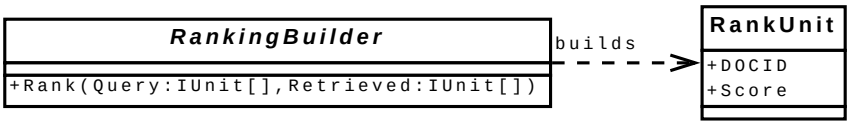


Figure A.16: Abstract class *RankingBuilder* and *RankUnit* type

use more than one inverted file at a single moment. However, this means that the process might require the inverted file to remain on the disk, which significantly affects the performance of the system. To improve such cases, a cache system is implemented, which improves the performance of the system since frequently used information is kept in main memory, and only the less frequently used information is accessed on the disk.

If speed performance needs to be improved, the retrieval process can also be distributed to different machines similarly to the indexing process. However, we have not yet implementing a strategy that will distribute and merge results from different machines and inverted files.

A.3 Jemu System

Jemu is an IR system that was developed using the Tupi Framework. The Jemu system was used for all comparative experiments of IR techniques described in this thesis, in particular those in Chapter 6. Jemu was developed so that with just a simple change of parameters read in at runtime, it could access and retrieve different types of relational information. Jemu was also developed so that the same IR methodologies can be applied to different types of information. This allows us to run experiments that highlight the differences given by the information type, and not by the technique used.

A.3.1 Implementation

The Jemu system is made of two main programs: the *JemuIndexer* and the *JemuRetriever*. These programs aggregate four types of information: BoW, syntactic dependencies, QPLM, and PropBank-style SRL. A simplified version of Jemu's architecture is shown in Figure A.17. Both the implementation

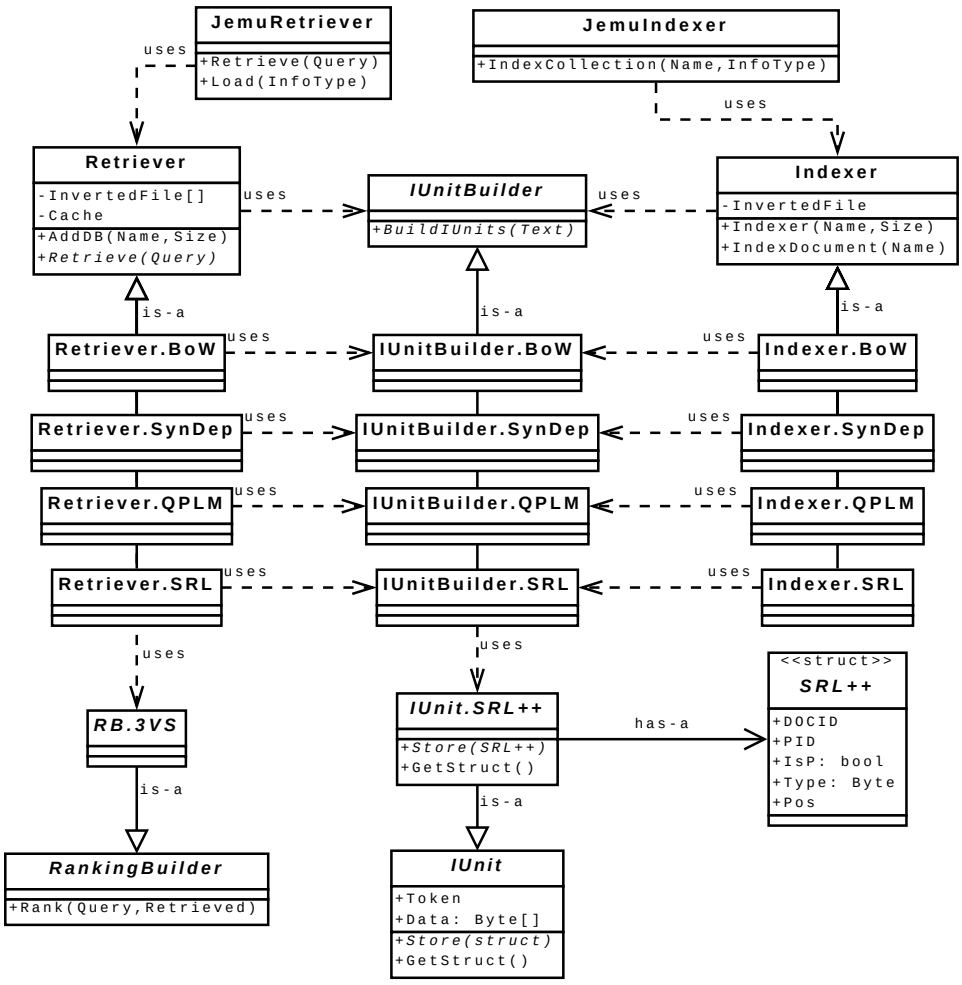


Figure A.17: Simplified overview of the Jemu architecture

of *JemuIndexer* and *JemuRetriever* require the implementation of specialised *IUnitBuilder* and *IUnit* for each information type. The implementations of *Indexer* and *Retriever* for each information type are trivial, since they only require the use of the respective *IUnitBuilder* and *IUnit*.

The implementation of the *JemuRetriever* required the definition of a ranking algorithm in *RankingBuilder*. The implementation of the *RankingBuilder* corresponds to the extension of the vector space ranking presented in Chapter 4 on page 104.

A.3.2 Implementation of *IUnit*

When implementing the *IUnit* structure, we noticed that could fit all four different types of information that we use into the PropBank's SRL definition. Thus, we created only one specialised *IUnit* structure called *IUnit.SRL++* (see Figure A.17).

Using a more complex structure to store simpler information is not an issue in the multi-layer inverted file representation (see Chapter 4). Our representation optimises the space allocation when fields in *IUnit.SRL++* are not used, or when they are repeated in a list of *IUnit.SRL++*.

When using *IUnit.SRL++* to store PropBank-style SRL, the following mapping takes place:

- *dDID* receives a document ID;
- *dPNO* receives a predicate ID (unique to a document);
- *bIsP* receives *true* for predicates and *false* for arguments;
- *cTYPE* receives an unique value according to its argument type (e.g. arg0, arg1 and arg2), or whether it is a predicate; and
- *dPOS* is the position within a particular argument or predicate.

Table A.1 illustrates the representation of the sentence "*John Smith loves Mary Jones*" when semantically annotated with PropBank-style semantic roles and encoded as *IUnit.SRL++*. The example of Table A.1 uses arbitrarily defined Document ID and dPNO, and the following semantic representation:

$$[_{\text{arg0}} \text{ John Smith}] \; [_{\text{pred}} \; \textbf{loves}] \; [_{\text{arg1}} \text{ Mary Jones}]$$

In order to represent the other types of information we apply the mapping shown in Figure A.2.

Objects of the class *IUnit.SRL++* are built using a specific *IUnitBuilder* for each of the information types used.

IUnit.SRL++	**John**	**Smith**	**loves**	**Mary**	**Jones**
dDID	1002	1002	1002	1002	1002
dPNO	15	15	15	15	15
bIsP	*false*	*false*	*true*	*false*	*false*
cTYPE	'0'	'0'	'P'	'1'	'1'
dPOS	1	2	1	1	2

Table A.1: *IUnit.SRL++* representations for the sentence "*John Smith loves Mary Jones*"

	BoW	**syntactic dependencies**	**QPLM**
dDID	document ID	document ID	document ID
dPNO	paragraph ID	syntactic dependency ID	sentence ID
bIsP	*false* for every term	*true* for the head of the dependency and *false* for the modifier	*true* for the $\mathcal{W}$ part of the QPLM and *false* for $\mathcal{A}$ part
cTYPE	0 for every term	code indicating the part of speech	code indicating the type of QPLM relation
dPOS	position of the word in the paragraph	code indicating the type of syntactic dependency	QPLM triple ID

Table A.2: Mapping of BoW, syntactic dependencies and QPLM to the structure of *IUnit.SRL++*

A.3.3 Implementation of *IUnitBuilder*

The *IUnitBuilder* was implemented as four specialised classes as shown in Figure A.17. All *IUnitBuilder* create *IUnit.SRL++* in different ways according the type of information that has been built.

The *IUnitBuilder.BoW* finds the terms for *IUnit.SRL++* using standard word boundaries such as spaces, punctuation marks and dashes. The *IUnitBuilder.Syn* builds *IUnit.SRL++* with the help of the Connexor syntactic parser [143]. The *IUnitBuilder.QPLM* builds *IUnit.SRL++* using the QPLM annotation tool described in Chapter 5. The *IUnitBuilder.SRL* builds *IUnit.SRL++* with the help of the SwiRL [142].

A.3.4 Implementation of *RankingBuilder*

The module *RankingBuilder* is implemented so that the function *Rank* can build a ordered list of documents. The function receives the query as a list of *IUnit*s and all *IUnit*s that were retrieved in the Inverted File for the given query.

We implemented one *RankingBuilder* called *RB.3VS* for triple-vector space (as defined in Chapter 4 on page 104). The relations among *IUnit*s are established in *RankingBuilder.3VS*. This means that the performance of the ranking function is bound to the size of the corpus and the frequency of occurrence of the query words.

A.4 Concluding Remarks

We have implemented the Tupi framework and the Jemu system in a standard environment to analyse the impact of the use of different linguistic information in IR. The Tupi framework allows the development of IR systems that use different types of relational information with minimal effort. Using the Tupi framework, we have implemented the Jemu system, which only required the implementation of a few modules.

B

Question Sets

B.1 List of Trustable Questions

The following questions make the list of trustable questions FACTOID-TRUSTED defined in Chapter 6 on page 145. The list is ordered by question ID and contains a *Topic*, a *Question* and an *Answer pattern*.[1] The remaining columns indicate the number of documents that contain the answer pattern (*Doc*), the number of documents that are correctly answered by the pattern (*OK*), and the probability that a document that matches the answer pattern correctly answers the question (*Prob*).[2] The questions that form question set FACTOID-1A are indicated by a symbol † before the question ID.

ID	Topic	Question	Answer pattern	Doc	OK	Prob
5.4	AARP	Who is its top official or CEO?	((Deets)|(Tess Canja))	40	38	95%
13.2	Jar Jar Binks	What actor is used as his voice?	Ahmed Best	27	25	93%
† 16.1	cataract	What is the primary symptom of a cataract?	clouding of the lens in the eye	1	1	100%
† 18.2	boxer Floyd Patterson	When did he win the title?	Nov. 30, 1956	1	1	100%
† 19.4	Kibbutz	Where was the first Kibbutz founded?	shores of the Sea of Galiee	1	1	100%
† 26.2	Ice-T	What was his original name?	Tracy Morrow	1	1	100%
† 30.1	minstrel Al Jolson	What is Al Jolson's real name?	Asa Yoelson	1	1	100%
† 33.4	Florence Nightingale	When did she die?	Aug(\.?|ust) (_|\s)13\s?, (_|\s)1910	1	1	100%
34.4	Amtrak	Who is the president or chief executive of Amtrak?	George (D.)?Warrington	35	33	94%

[1] As made available by TREC in `http://trec.nist.gov/data/qamain.html`

[2] See Equation (6.1) on page 144

ID	Topic	Question	Answer pattern	Doc	OK	Prob
36.3	Khmer Rouge	Who was its first leader?	Saloth Sar	4	4	100%
46.1	Heaven's Gate	Who was the leader of the cult?	Marshall Herff Applewhite	3	3	100%
† 46.6	Heaven's Gate	Why did they commit suicide?	to join a flying saucer hiding behind the Hale-Bopp comet	1	1	100%
48.2	Abu Nidal	What was his name at birth?	Sabri (a|e)l(\-|\s)Banna	25	23	92%
49.3	Carlos the Jackal	Whom did he marry?	Magdalena Kopp	2	2	100%
53.1	Conde Nast	Who is the CEO of the publishing company Conde Nast?	Steven (T.)?Florio	34	31	91%
61.1	Muslim Brotherhood	Who founded the Muslim Brotherhood?	Hassan (e|a)l(\-|\s)Banna	2	2	100%
† 62.3	Berkman Center for Internet and Society	What is its mission?	studies the development and regulation of the Internet	1	1	100%
66.2	Russian submarine Kursk sinks	Who was the on-board commander of the submarine?	(Capt(.?|ain))?(Gennady)?Lyachin	16	18	100%
† 72.3	Bollywood	What is the Bollywood equivalent of Beverly Hills?	Malabar Hill	1	1	100%
73.4	Viagra	What is the scientific name for Viagra?	(S|s)ildenafil citrate	10	10	100%
75.4	Merck & Co.	What is the company's web address?	www.merck.com	5	5	100%
† 75.6	Merck & Co.	Who was a chairman of the company in 1996?	(Raymond (V.?)?)?Gilmarman	1	1	100%
80.2	Crash of EgyptAir Flight 990	Who was the pilot of Flight 990?	(Ahmed(Mahmoud)?)?(el-|El |al)Habash(y|i)	34	32	94%
† 81.5	Preakness 1998	What was the track attendance for the 1998 Preakness?	103,269	1	1	100%
82.2	Howdy Doody Show	On what date did the show go off the air?	(Sept.?|September) 30:? 1960	3	3	100%
84.3	meteorites	What is it called by the Indians?	Tomanoas	3	3	100%
† 85.3	Norwegian Cruise Lines (NCL)	What is the name of the NCL's own private island?	Great Stirrup Cay	1	1	100%
† 85.5	Norwegian Cruise Lines (NCL)	Why did the Grand Cayman turn away a NCL ship?	chartered for gay passengers	1	1	100%
89.6	Little League Baseball	What is Little League Baseball's URL on the Internet?	(http://)? www.littleleague.org	7	7	100%
91.1	Cliffs Notes	Who originated Cliffs Notes?	(Cliff)?Hillegass	10	10	100%
† 96.1	1998 Nagano Olympic Games	What materials was the 1998 Olympic torch made of?	pine and bamboo	1	1	100%
† 96.2	1998 Nagano Olympic Games	How long was the men's downhill ski run in Nagano?	2:923 meters	1	1	100%
† 98.4	American Legion	What organization has helped to revitalize Legion membership?	Sons of the Legion	1	1	100%
99.3	Woody Guthrie	Where was Guthrie born?	Okemah(:? (Okla.?|Oklahoma))?	6	6	100%
105.5	1980 Mount St. Helens eruption	What was the height of the volcano after the eruption?	8:363 feet	2	2	100%
109.4	Telefonica of Spain	Name the chairman.	(Juan)?Villalonga	50	45	90%
126.2	Pope Pius XII	What was his name before becoming Pope?	(Eugenio)?Pacelli	5	5	100%

ID	Topic	Question	Answer pattern	Doc	OK	Prob
† 126.4	Pope Pius XII	How long was his pontificate?	19(-|)years?(:? seven months and seven days)	1	1	100%
† 132.5	Kim Jong Il	To whom is Kim Jong Il married?	Kim Young Sook	1	1	100%
† 144.4	82nd Airborne Division	What is the division's motto?	shoot 'em in the face	1	1	100%
148.4	tourists massacred at Luxor in 1997	At what site in Luxor did the massacre occur?	(Hatshepsut Temple|Temple of Hatshepsut)	30	28	93%
149.6	The Daily Show	What was the title for The Daily Show's 2000 election coverage?	Indecision 2000	15	14	93%
† 152.2	Wolfgang Amadeus Mozart	When was Mozart born?	Jan(.|uary) 27 ?: 1756	1	1	100%
153.1	Alfred Hitchcock	When was Hitchcock born?	(Aug(.|ust).*13 .*1899|1899-08-13)	0	4	100%
153.6	Alfred Hitchcock	When did Hitchcock die?	April 29, 1980	2	2	100%
† 165.6	the Queen Mum's 100th Birthday	What was her husband's title when she married him?	Prince Albert: Duke of York	1	1	100%
166.5	avian flu outbreak in Hong Kong	What new strain of avian flu appeared in Hong Kong within the next two years?	H9N2	2	2	100%
† 168.6	Prince Charles' paintings	What other prince showed his paintings in a two-prince exhibition with Prince Charles in London?	Khalid Al-Faisal	1	1	100%
† 173.3	World Tourism Organization (WTO)	Who was the director of the WTO in 1996?	Dr. Eduardo Fayos-Sola	1	1	100%
† 174.5	American Farm Bureau Federation (AFBF)	According to the AFBF, what is the average cost of a turkey dinner with all the trimmings?	$33. ?83	1	1	100%
† 178.5	methamphetamine labs	What chemical company's labels were found on bottles in a methamphetamine lab?	Chemco International	1	1	100%
179.1	Hedy Lamarr	What was the date of Hedy Lamarr's birth?	Nov. 9: 1913	2	2	100%
179.2	Hedy Lamarr	What was her name at birth?	Hedwig Eva Marie Kiesler	2	2	100%
† 184.6	1999 Chicago Marathon	What was the women's winning time in the 1999 Chicago Marathon?	two hours 25 minutes and 59 seconds	1	1	100%
185.4	Iditarod Race	Who is the founder of the Iditarod?	Joe Redington	4	4	100%
185.6	Iditarod Race	How many miles long is the Iditarod?	1.*049.*miles?	0	4	100%
† 187.6	Amazon River	What is the name of the Amazon River at its origin?	Carhuasanta River	1	1	100%
202.7	Cole Porter	What was Cole Porter's first professional musical?	See America First	2	2	100%
203.2	Nissan Corp.	Who is president of Nissan Corp.?	Carlos Ghosn	66	64	97%
† 206.3	Johnstown flood	How much water fell on Johnstown?	(20-million(-|)ton|20 - 1000000 - ton)	1	1	100%

B.2 List of Self-Contained Questions

The following list of questions were automatically generated from the TREC questions in a way to create a self-contained query that can be used in a single stage retrieval. The process used to create this question is described in Chapter 6 on page 146.

ID		ID	
1.1	Regarding Crips, When was the first Crip gang started?	1.2	Regarding Crips, What does the name mean or come from?
1.3	Regarding Crips, Which cities have Crip gangs?	1.4	Regarding Crips, What ethnic group/race are Crip members?
1.5	What is Crips gang color?	1.6	Tell me more about Crips.
2.1	What is the name of Fred Durst's group?	2.2	What record company is Fred Durst with?
2.3	What are titles of Fred Durst's releases?	2.4	Where was Fred Durst born?
2.5	Tell me more about Fred Durst.	3.1	When was the Hale Bopp comet discovered?
3.2	How often does Hale Bopp comet approach the earth?	3.3	In what countries was the Hale Bopp comet visible on its last return?
3.4	Tell me more about Hale Bopp comet.	4.1	When was James Dean born?
4.2	When did James Dean die?	4.3	How did James Dean die?
4.4	What movies did James Dean appear in?	4.5	Which was the first movie that James Dean was in?
4.6	Tell me more about James Dean.	5.1	What does AARP stand for?
5.2	When was AARP started?	5.3	Where is AARP headquarters?
5.4	Who is AARP top official or CEO?	5.5	What companies has AARP endorsed?
5.6	Tell me more about AARP.	6.1	How long does one study as a Rhodes scholars scholar?
6.2	Where do Rhodes scholars study?	6.3	Name famous people who have been Rhodes scholars.
6.4	What countries have Rhodes scholars come from?	6.5	Tell me more about Rhodes scholars.
7.1	What kind of animal is an agouti?	7.2	What is agouti average life span?
7.3	In what countries are agouti found?	7.4	Tell me more about agouti.
8.1	Who founded the Black Panthers organization?	8.2	When was Black Panthers founded?
8.3	Where was Black Panthers founded?	8.4	Who have been members of Black Panthers?
8.5	Tell me more about Black Panthers.	9.1	Who are the members of Insane Clown Posse?
9.2	What albums have Insane Clown Posse made?	9.3	What is Insane Clown Posse style of music?
9.4	What is Insane Clown Posse biggest hit?	9.5	Tell me more about Insane Clown Posse.
10.1	What are prions made of?	10.2	Who discovered prions?
10.3	What diseases are prions associated with?	10.4	What researchers have worked with prions?
10.5	Tell me more about prions.	11.1	Who is the lead singer/musician in the band Nirvana?
11.2	Who are the the band Nirvana members?	11.3	When was the the band Nirvana formed?
11.4	What is the band Nirvana biggest hit?	11.5	What are the band Nirvana albums?
11.6	What style of music do the band Nirvana play?	11.7	Tell me more about the band Nirvana.
12.1	What industry is Rohm and Haas in?	12.2	Where is Rohm and Haas located?
12.3	What is Rohm and Haas annual revenue?	12.4	How many employees does Rohm and Haas have?
12.5	Tell me more about Rohm and Haas.	13.1	What film introduced Jar Jar Binks?
13.2	What actor is used as Jar Jar Binks voice?	13.3	To what alien race does Jar Jar Binks belong?
13.4	Tell me more about Jar Jar Binks.	14.1	Horus is the god of what?
14.2	What country is Horus associated with?	14.3	Who was Horus mother?
14.4	Who was Horus father?	14.5	Tell me more about Horus.
15.1	Who are the members of the Rat Pack?	15.2	Regarding Rat Pack, Who coined the name?
15.3	What Las Vegas hotel was made famous by the Rat Pack?	15.4	Tell me more about Rat Pack.
16.1	What is the primary symptom of a cataract?	16.2	How are cataract treated?
16.3	Who are doctors that have performed cataract surgery?	16.4	Tell me more about cataract.

ID		ID	
17.1	When was the International Criminal Court established?	17.2	What kind of cases does International Criminal Court try?
17.3	Who is the sponsor of the International Criminal Court?	17.4	How many justices are members of the International Criminal Court?
17.5	Tell me more about International Criminal Court.	18.1	What division (weight) did boxer Floyd Patterson win?
18.2	When did boxer Floyd Patterson win the title?	18.3	How old was boxer Floyd Patterson when he won the title?
18.4	Who did boxer Floyd Patterson beat to win the title?	18.5	Who beat boxer Floyd Patterson to take the title away?
18.6	List the names of boxer Floyd Pattersons he fought?	18.7	Tell me more about boxer Floyd Patterson.
19.1	What kind of a community is a Kibbutz?	19.2	With what country are Kibbutz associated?
19.3	When was the first Kibbutz founded?	19.4	Where was the first Kibbutz founded?
19.5	Regarding Kibbutz, How many are there now?	19.6	Tell me more about Kibbutz.
20.1	In what year did the first Concorde passenger flight take place?	20.2	What airlines have Concordes in their fleets?
20.3	How many seats are in the cabin of a Concorde?	20.4	How fast does the Concorde fly?
20.5	What year was the first Concorde crash?	20.6	Tell me more about Concorde.
21.1	How many Club Med vacation spots are there worldwide?	21.2	Regarding Club Med, List the spots in the United States?
21.3	Where is an adults-only Club Med?	21.4	Tell me more about Club Med.
22.1	Where was Franz Kafka born?	22.2	When was Franz Kafka born?
22.3	What is Franz Kafka ethnic background?	22.4	What books did Franz Kafka author?
22.5	Tell me more about Franz Kafka.	23.1	In what film is Gordon Gekko the main character?
23.2	Regarding Gordon Gekko, Who plays the role?	23.3	Regarding Gordon Gekko, What year was the movie released?
23.4	What was Gordon Gekko's profession?	23.5	Tell me more about Gordon Gekko.
24.1	When was architect Frank Gehry born?	24.2	What nationality is architect Frank Gehry?
24.3	Where did he study architect Frank Gehryure?	24.4	What prizes or awards has architect Frank Gehry won?
24.5	What buildings has architect Frank Gehry designed?	24.6	Tell me more about architect Frank Gehry.
25.1	What sport do Harlem Globe Trotters play?	25.2	When were Harlem Globe Trotters founded?
25.3	By whom were Harlem Globe Trotters founded?	25.4	What countries have Harlem Globe Trotters played in?
25.5	Tell me more about Harlem Globe Trotters.	26.1	What kind of singer is Ice-T?
26.2	What was Ice-T original name?	26.3	When was Ice-T born?
26.4	Where was Ice-T born?	26.5	What are names of Ice-T albums?
26.6	Tell me more about Ice-T.	27.1	What sport does Jennifer Capriati play?
27.2	Who is Jennifer Capriati coach?	27.3	Where does Jennifer Capriati live?
27.4	When was Jennifer Capriati born?	27.5	Tell me more about Jennifer Capriati.
28.1	What kind of business is Abercrombie and Fitch?	28.2	When was Abercrombie and Fitch established?
28.3	Where was Abercrombie and Fitch established?	28.4	Regarding Abercrombie and Fitch, How many stores are there?
28.5	Tell me more about Abercrombie and Fitch.	29.1	Why is the 'Tale of Genji' famous?
29.2	Who wrote Tale of Genji?	29.3	When was Tale of Genji written?
29.4	Tell me more about Tale of Genji.	30.1	What is minstrel Al Jolson Jolson's real name?
30.2	What is minstrel Al Jolson nationality?	30.3	Where was minstrel Al Jolson born?
30.4	Who did minstrel Al Jolson marry?	30.5	What songs did minstrel Al Jolson sing?
30.6	Tell me more about minstrel Al Jolson.	31.1	What was Jean Harlow's real name?
31.2	Where was Jean Harlow born?	31.3	When did Jean Harlow die?
31.4	How old was Jean Harlow when she died?	31.5	What did Jean Harlow die of?
31.6	Where is Jean Harlow buried?	31.7	What movies did Jean Harlow appear in?
31.8	What leading men did Jean Harlow star opposite of?	31.9	Tell me more about Jean Harlow.
32.1	What do practitioners of Wicca worship?	32.2	How many followers does Wicca have?
32.3	Who is Wicca leader?	32.4	What festivals does Wicca have?
32.5	Tell me more about Wicca.	33.1	What is Florence Nightingale famous for?
33.2	When was Florence Nightingale born?	33.3	Where was Florence Nightingale born?
33.4	When did Florence Nightingale die?	33.5	Tell me more about Florence Nightingale.
34.1	When did Amtrak begin operations?	34.2	How many passengers does Amtrak serve annually?
34.3	How many employees does Amtrak have?	34.4	Who is the president or chief executive of Amtrak?

ID		ID	
34.5	Name cities that have an Amtrak terminal.	34.6	Tell me more about Amtrak.
35.1	When did Jack Welch become chairman of General Electric?	35.2	How many years was Jack Welch with GE?
35.3	When did Jack Welch retire from GE?	35.4	How many people did Jack Welch fire from GE?
35.5	Tell me more about Jack Welch.	36.1	Regarding Khmer Rouge, In what country did this movement take place?
36.2	When did the Khmer Rouge come into power?	36.3	Who was Khmer Rouge first leader?
36.4	Who were leaders of the Khmer Rouge?	36.5	When was the Khmer Rouge removed from power?
36.6	Tell me more about Khmer Rouge.	37.1	How many members are there in the singing group the Wiggles?
37.2	Regarding Wiggles, Who are the members' names?	37.3	Where is Wiggles from?
37.4	List the Wiggles' songs.	37.5	Tell me more about Wiggles.
38.1	Regarding quarks, What kind of a particle is a quark?	38.2	Who discovered quarks?
38.3	When were quarks discovered?	38.4	What are the different types of quarks?
38.5	Tell me more about quarks.	39.1	Regarding The Clash, What kind of music does the band play?
39.2	In what year was The Clash first major album recorded?	39.3	Name The Clash songs?
39.4	Tell me more about The Clash.	40.1	When was Chester Nimitz born?
40.2	What town was Chester Nimitz native of?	40.3	What branch of the military did Chester Nimitz serve in?
40.4	During what war did Chester Nimitz serve?	40.5	What rank did Chester Nimitz reach?
40.6	Tell me more about Chester Nimitz.	41.1	What year did the Teapot Dome scandal take place?
41.2	Regarding Teapot Dome scandal, Who was President of the United States at the time?	41.3	What was the main focus of the Teapot Dome scandal?
41.4	Who were the major players involved in the Teapot Dome scandal?	41.5	Tell me more about Teapot Dome scandal.
42.1	When was the USS Constitution commissioned?	42.2	What conflict did USS Constitution distinguish herself in?
42.3	How many battles did USS Constitution win?	42.4	What is USS Constitution nickname?
42.5	Tell me more about USS Constitution.	43.1	Regarding Nobel prize, Who established the awards?
43.2	What are the different categories of Nobel prizes?	43.3	Regarding Nobel prize, When were the awards first given?
43.4	What is the monetary value of the Nobel prize?	43.5	Tell me more about Nobel prize.
44.1	What tribe did Sacajawea belong to?	44.2	What years did Sacajawea accompany Lewis and Clark on their expedition?
44.3	When was Sacajawea born?	44.4	When did Sacajawea die?
44.5	Where is Sacajawea buried?	44.6	How much is the Sacajawea coin worth?
44.7	Tell me more about Sacajawea.	45.1	When was the International Finance Corporation (IFC) established?
45.2	What is International Finance Corporation (IFC) mission?	45.3	What countries has the IFC International Finance Corporation (IFC)d projects in?
45.4	Tell me more about International Finance Corporation (IFC).	46.1	Regarding Heaven's Gate, Who was the leader of the cult?
46.2	How many of its members committed Heaven's Gateuicide?	46.3	When did the mass Heaven's Gateuicide occur?
46.4	Where did Heaven's Gate occur?	46.5	How did they commit Heaven's Gateuicide?
46.6	Why did they commit Heaven's Gateuicide?	46.7	Tell me more about Heaven's Gate.
47.1	On what date was Bashar Assad inaugurated as the Syrian president?	47.2	What is Bashar Assad party affiliation?
47.3	What was Bashar Assad profession prior to assuming the Presidency?	47.4	Regarding Bashar Assad, How long are Syrian presidential terms?
47.5	What schools did Bashar Assad attend?	47.6	Tell me more about Bashar Assad.
48.1	When was Abu Nidal born?	48.2	What was Abu Nidal name at birth?
48.3	How many followers does Abu Nidal have?	48.4	In what countries has Abu Nidal operated from?
48.5	In what year did the PLO condemn Abu Nidal to death?	48.6	Tell me more about Abu Nidal.
49.1	What is Carlos the Jackal' real name?	49.2	Where was Carlos the Jackal born?
49.3	Whom did Carlos the Jackal marry?	49.4	When was Carlos the Jackal captured?

ID		ID	
49.5	Where was Carlos the Jackal captured?	49.6	Tell me more about Carlos the Jackal.
50.1	When was Cassini space probe launched?	50.2	How much did Cassini space probe cost to build?
50.3	What is Cassini space probe destination?	50.4	What planets will Cassini space probe pass?
50.5	Tell me more about Cassini space probe.	51.1	What is the religious affiliation of the Kurds?
51.2	How many Kurds live in Turkey?	51.3	What other countries do Kurds live in?
51.4	Tell me more about Kurds.	52.1	When was the first Burger King restaurant opened?
52.2	Regarding Burger King, Where was the first restaurant opened?	52.3	Who founded Burger King?
52.4	What are Burger King's gross sales today?	52.5	What countries is Burger King located in?
52.6	Tell me more about Burger King.	53.1	Who is the CEO of the publishing company Conde Nast?
53.2	Where are Conde Nast's headquarters?	53.3	When was Conde Nast founded?
53.4	What magazines does Conde Nast publish?	53.5	Tell me more about Conde Nast.
54.1	Where was Eileen Marie Collins born?	54.2	How many siblings does Eileen Marie Collins have?
54.3	Whom did Eileen Marie Collins marry?	54.4	How many children does Eileen Marie Collins have?
54.5	Where does Eileen Marie Collins live?	54.6	What schools did Eileen Marie Collins attend?
54.7	What is Eileen Marie Collins occupation?	54.8	What branch of the service did Eileen Marie Collins serve in?
54.9	What rank did Eileen Marie Collins reach?	54.10	Tell me more about Eileen Marie Collins.
55.1	Where was Walter Mosley born?	55.2	When was Walter Mosley born?
55.3	How old was Walter Mosley when he began writing?	55.4	What books has Walter Mosley written?
55.5	Tell me more about Walter Mosley.	56.1	When was the Good Friday Agreement made?
56.2	What is the purpose of the Good Friday Agreement?	56.3	What groups are affected by Good Friday Agreement?
56.4	Who were the key players in negotiating the Good Friday Agreement?	56.5	Tell me more about Good Friday Agreement.
57.1	What kind of ship is the Liberty Bell 7?	57.2	Who developed Liberty Bell 7?
57.3	What is Liberty Bell 7 made of?	57.4	What happened to Liberty Bell 7?
57.5	Tell me more about Liberty Bell 7.	58.1	What is philanthropist Alberto Vilar's nationality?
58.2	What organizations has philanthropist Alberto Vilar donated money to?	58.3	What is the name of philanthropist Alberto Vilar he founded?
58.4	What companies has philanthropist Alberto Vilar invested in?	58.5	Tell me more about philanthropist Alberto Vilar.
59.1	Who founded Public Citizen?	59.2	When was Public Citizen formed?
59.3	What is Public Citizen purpose?	59.4	How many members does Public Citizen have?
59.5	Who is Public Citizen current head?	59.6	Tell me more about Public Citizen.
60.1	What state does senator Jim Inhofe represent?	60.2	When was senator Jim Inhofe born?
60.3	When was senator Jim Inhofe first elected to the senate?	60.4	What branch of the service did senator Jim Inhofe serve in?
60.5	Tell me more about senator Jim Inhofe.	61.1	Who founded the Muslim Brotherhood?
61.2	When was Muslim Brotherhood formed?	61.3	What is Muslim Brotherhood goal?
61.4	What countries does Muslim Brotherhood operate in?	61.5	Name members of Muslim Brotherhood?
61.6	Tell me more about Muslim Brotherhood.	62.1	Where is the Berkman Center for Internet and Society located?
62.2	When was the Berkman Center for Internet and Society formed?	62.3	What is Berkman Center for Internet and Society mission?
62.4	Name members of the Berkman Center for Internet and Society?	62.5	Tell me more about Berkman Center for Internet and Society.
63.1	What kind of insect is a boll weevil?	63.2	What type of plant does boll weevil damage?
63.3	What states have had problems with boll weevils?	63.4	Tell me more about boll weevil.
64.1	What was Johnny Appleseed's real name?	64.2	Where was Johnny Appleseed born?
64.3	When did Johnny Appleseed die?	64.4	What did Johnny Appleseed wear as a hat?
64.5	In what states did Johnny Appleseed plant trees?	64.6	Tell me more about Johnny Appleseed.
65.1	What are the names of the space shuttles?	65.2	Regarding space shuttles, Which was the first shuttle?

ID		ID	
65.3	Regarding space shuttles, When was the first flight?	65.4	When was the Challenger space shuttles shuttle disaster?
65.5	Regarding space shuttles, How many members were in the crew of the Challenger?	65.6	How long did the Challenger flight last before space shuttles exploded?
65.7	Tell me more about space shuttles.	66.1	When did the Russian submarine Kursk sinks sink?
66.2	Who was the on-board commander of the Russian submarine Kursk sinks?	66.3	The submarine was part of which Russian submarine Kursk sinks fleet?
66.4	Regarding Russian submarine Kursk sinks, How many crewmen were lost in the disaster?	66.5	Regarding Russian submarine Kursk sinks, Which countries expressed regret about the loss?
66.6	In what sea did the Russian submarine Kursk sinks sink?	66.7	Which U.S. Russian submarine Kursk sinkss were reportedly in the area?
66.8	Tell me more about Russian submarine Kursk sinks.	67.1	Regarding Miss Universe 2000 crowned, Who won the crown?
67.2	Regarding Miss Universe 2000 crowned, What country did the winner represent?	67.3	Regarding Miss Universe 2000 crowned, How many competitors did the winner have?
67.4	Regarding Miss Universe 2000 crowned, Where was the contest held?	67.5	Regarding Miss Universe 2000 crowned, What was the scheduled date of the contest?
67.6	Regarding Miss Universe 2000 crowned, Name other contestants?	67.7	Tell me more about Miss Universe 2000 crowned.
68.1	Where is Port Arthur Massacre Arthur?	68.2	When did the Port Arthur Massacre occur?
68.3	What was the final death toll of the Port Arthur Massacre?	68.4	Regarding Port Arthur Massacre, Who was the killer?
68.5	Regarding Port Arthur Massacre, What was the killer's nationality?	68.6	Regarding Port Arthur Massacre, What were the names of the victims?
68.7	Regarding Port Arthur Massacre, What were the nationalities of the victims?	68.8	Tell me more about Port Arthur Massacre.
69.1	When did France wins World Cup in soccer win the World Cup?	69.2	Who did France wins World Cup in soccer beat for the World Cup?
69.3	Regarding France wins World Cup in soccer, What was the final score?	69.4	Regarding France wins World Cup in soccer, What was the nickname for the French team?
69.5	Regarding France wins World Cup in soccer, At what stadium was the game played?	69.6	Regarding France wins World Cup in soccer, Who was the coach of the French team?
69.7	Regarding France wins World Cup in soccer, Name players on the French team?	69.8	Tell me more about France wins World Cup in soccer.
70.1	Regarding Plane clips cable wires in Italian resort, When did the accident occur?	70.2	Regarding Plane clips cable wires in Italian resort, Where in Italy did the accident occur?
70.3	Regarding Plane clips cable wires in Italian resort, How many people were killed?	70.4	What was the affiliation of the Plane clips cable wires in Italian resort?
70.5	Regarding Plane clips cable wires in Italian resort, What was the name of the pilot?	70.6	Regarding Plane clips cable wires in Italian resort, What was the outcome of the U.S. trial against the pilot?
70.7	Regarding Plane clips cable wires in Italian resort, Who were on-ground witnesses to the accident?	70.8	Tell me more about Plane clips cable wires in Italian resort.
71.1	What type of plane is an F16?	71.2	How fast can F16 fly?
71.3	Who manufactures the F16?	71.4	Where is F16 based?
71.5	Who manufactures engines for the F16?	71.6	What countries besides U.S. fly F16s?
71.7	Tell me more about F16.	72.1	Where is Bollywood located?
72.2	From what foreign city did Bollywood derive its name?	72.3	What is the Bollywood equivalent of Beverly Hills?
72.4	What is Bollywood's equivalent of the Oscars?	72.5	Where does Bollywood rank in the world's film industries?
72.6	Who are some of the Bollywood stars?	72.7	Tell me more about Bollywood.
73.1	Viagra is prescribed for what problem?	73.2	Who manufactures Viagra?
73.3	Who approved Viagra use in China?	73.4	What is the scientific name for Viagra?
73.5	When did Viagra go on the market?	73.6	In what countries could Viagra be obtained on the black market?
73.7	Tell me more about Viagra.	74.1	What type of school is DePauw University?
74.2	Where is DePauw University located?	74.3	When was DePauw University founded?
74.4	Who was president of DePauw University in 1999?	74.5	Regarding DePauw University, What was the approximate number of students attending in 1999?
74.6	Name graduates of the DePauw University?	74.7	Tell me more about DePauw University.
75.1	Where is the Merck & Company headquartered?	75.2	What does the Merck & Company make?

ID		ID	
75.3	What is Merck & Co. symbol on the New York Stock Exchange?	75.4	What is the Merck & Company's web address?
75.5	Name Merck & Companies that are business competitors?	75.6	Who was a chairman of the Merck & Company in 1996?
75.7	Name products manufactured by Merck & Co.?	75.8	Tell me more about Merck & Co..
76.1	What was Bing Crosby profession?	76.2	For which movie did Bing Crosby win an Academy Award?
76.3	What was Bing Crosby nickname?	76.4	What is the title of Bing Crosby all-time best-selling record?
76.5	He is an alumnus of which university? Bing Crosby is an alumnus of which university?	76.6	How old was Bing Crosby when he died?
76.7	What movies was Bing Crosby in?	76.8	Tell me more about Bing Crosby.
77.1	When was George Foreman born?	77.2	Where was George Foreman born?
77.3	When did George Foreman first become world heavyweight boxing champion?	77.4	Who did George Foreman defeat for his first heavyweight championship?
77.5	How old was George Foreman when he first won the heavyweight championship?	77.6	Name opponents who George Foreman defeated?
77.7	Name opponents who defeated George Foreman?	77.8	Tell me more about George Foreman.
78.1	When did Akira Kurosawa die?	78.2	When was Akira Kurosawa born?
78.3	Which university did Akira Kurosawa graduate from?	78.4	What was Akira Kurosawa profession?
78.5	What was Akira Kurosawa English nickname?	78.6	What was Akira Kurosawa wife's profession?
78.7	What were some of Akira Kurosawa Japanese film titles?	78.8	Tell me more about Akira Kurosawa.
79.1	When did the Kip Kinkel school shooting shooting occur?	79.2	Regarding Kip Kinkel school shooting, How many students were wounded?
79.3	List students who were shot by Kip Kinkel school shooting Kinkel?	79.4	How many students did Kip Kinkel school shooting kill?
79.5	How old was Kip Kinkel school shooting Kinkel when the shooting took place?	79.6	How many bombs did investigators find in Kip Kinkel school shooting's home?
79.7	Tell me more about Kip Kinkel school shooting.	80.1	Where in the Atlantic Ocean did Flight 990 Crash of EgyptAir Flight 990?
80.2	Who was the pilot of Crash of EgyptAir Flight 990 990?	80.3	Who was the co-pilot of Crash of EgyptAir Flight 990 990?
80.4	Regarding Crash of EgyptAir Flight 990, How many crew members were aboard?	80.5	How many passengers were aboard Crash of EgyptAir Flight 990 990?
80.6	Identify the nationalities of passengers on Crash of EgyptAir Flight 990 990?	80.7	Tell me more about Crash of EgyptAir Flight 990.
81.1	Name the horse that won the Preakness 1998 in 1998?	81.2	List other horses who won the Kentucky Derby and Preakness 1998 but not the Belmont?
81.3	Who is the trainer of the Preakness 1998 winner?	81.4	Who finished second to the Preakness 1998 winner in 1998?
81.5	What was the track attendance for the 1998 Preakness 1998?	81.6	Regarding Preakness 1998, What time did the race begin?
81.7	Tell me more about Preakness 1998.	82.1	What year did the "Howdy Doody Show" first run on television?
82.2	On what date did the Howdy Doody Show go off the air?	82.3	Name the various puppets used in the "Howdy Doody Show".
82.4	Name the characters in the Howdy Doody Show?	82.5	Regarding Howdy Doody Show, The main puppet character was based on what person?
82.6	Tell me more about Howdy Doody Show.	83.1	What was the Louvre Museum before it was a museum?
83.2	When was the Louvre Museum transformed into a museum?	83.3	How many paintings are on permanent exhibit at the Louvre Museum?
83.4	Name the works of art that have been stolen from the Louvre Museum?	83.5	How many people visit the Louvre Museum each year?
83.6	Who is president/director of the Louvre Museum?	83.7	Tell me more about Louvre Museum.
84.1	Regarding meteorites, What is the largest meteorite found in the U.S?	84.2	How heavy is meteorites?
84.3	What is meteorites called by the Indians?	84.4	Regarding meteorites, Where is the world's largest meteorite?
84.5	Regarding meteorites, How heavy is the world's largest meteorite?	84.6	How many metric tons of meteorites fall to the earth each year?
84.7	Provide a list of names or identifications given to meteorites.	84.8	Tell me more about meteorites.

ID		ID	
85.1	Name the ships of the Norwegian Cruise Lines (NCL)?	85.2	What Norwegian Cruise Lines (NCL) line attempted to take over NCL in December 1999?
85.3	What is the name of the Norwegian Cruise Lines (NCL)'s own private island?	85.4	How does NCL rank in size with other Norwegian Cruise Lines (NCL) lines?
85.5	Why did the Grand Cayman turn away a Norwegian Cruise Lines (NCL) ship?	85.6	Name so-called theme Norwegian Cruise Lines (NCL)s promoted by NCL?
85.7	Tell me more about Norwegian Cruise Lines (NCL).	86.1	Give the month and year that General Sani Abacha had a successful coup in Nigeria?
86.2	What reportedly caused the death of Sani Abacha?	86.3	How old was Sani Abacha when he died?
86.4	Who was sworn in to replace Sani Abacha?	86.5	Name the children of Sani Abacha.
86.6	Tell me more about Sani Abacha.	87.1	When was Enrico Fermi born?
87.2	When did Enrico Fermi die?	87.3	What Nobel Prize was Enrico Fermi awarded in 1938?
87.4	List things named in honor of Enrico Fermi.	87.5	What is Enrico Fermi most known for?
87.6	Give the name and symbol for the chemical element named after Enrico Fermi.	87.7	What country did Enrico Fermi come from originally?
87.8	Tell me more about Enrico Fermi.	88.1	Where is United Parcel Service (UPS) headquarters located?
88.2	Who is the CEO of United Parcel Service (UPS)?	88.3	When was United Parcel Service (UPS)'s first public stock offering?
88.4	In what foreign countries does the United Parcel Service (UPS) operate?	88.5	What color are United Parcel Service (UPS) trucks?
88.6	How much money did United Parcel Service (UPS) pay out in insurance claims in 1984?	88.7	Tell me more about United Parcel Service (UPS).
89.1	Where is the Little League Baseball League World Championship played?	89.2	On what street are the fields where the Little League Baseball League World Series is played?
89.3	What Little League Baseball League teams have won the World Series?	89.4	How many girls have played in the Little League Baseball League World Series?
89.5	What year was the first Little League Baseball League World Series played?	89.6	What is Little League Baseball's URL on the Internet?
89.7	Tell me more about Little League Baseball.	90.1	What grape varieties are Virginia wines made from?
90.2	Approximately how many acres of grapes are grown in Virginia wine?	90.3	Which Virginia wine vineyard produces the most wine?
90.4	Who was Virginia wine's first and most famous wine maker?	90.5	Name the Virginia wine festivals.
90.6	Who was the former CEO who became a Virginia wine maker?	90.7	Tell me more about Virginia wine.
91.1	Who originated Cliffs Notes?	91.2	Whose works were the subject of the first Cliffs Notes?
91.3	Give the titles of Cliffs Notes Condensed Classics.	91.4	What company now owns Cliffs Notes?
91.5	How many copies of Cliffs Notes are sold annually?	91.6	What percentage of Americans have used Cliffs Notes?
91.7	Tell me more about Cliffs Notes.	92.1	How many times did Arnold Palmer win the Masters?
92.2	How many times did Arnold Palmer win the British Open?	92.3	What players has Arnold Palmer competed against in the Skins Games?
92.4	Which golf courses were designed by Arnold Palmer?	92.5	What major championship did Arnold Palmer never win?
92.6	What was Arnold Palmer's wife's first name?	92.7	Tell me more about Arnold Palmer.
93.1	Who moderated the first 2000 Bush-Gore presidential debate 2000 presidential debate?	93.2	How long was the first 2000 Bush-Gore presidential debate scheduled to be?
93.3	On what university campus was the first 2000 Bush-Gore presidential debate debate held?	93.4	Which major network decided not to televise the first 2000 Bush-Gore presidential debate?
93.5	In what state did Al Gore prepare for the first 2000 Bush-Gore presidential debate debate?	93.6	On what date was the first 2000 Bush-Gore presidential debate debate?
93.7	Regarding first 2000 Bush-Gore presidential debate, Who helped the candidates prepare?	93.8	Tell me more about first 2000 Bush-Gore presidential debate.
94.1	Who was Mrs. 1998 indictment and trial of Susan McDougal's lawyer?	94.2	Regarding 1998 indictment and trial of Susan McDougal, Who was the prosecutor?
94.3	How did Mrs. 1998 indictment and trial of Susan McDougal plead?	94.4	Who testified for Mrs. 1998 indictment and trial of Susan McDougal's defense?

ID		ID	
94.5	Regarding 1998 indictment and trial of Susan McDougal, What was the jury's ruling on the obstruction of justice charge?	94.6	Regarding 1998 indictment and trial of Susan McDougal, What was the result of the contempt charges?
94.7	Tell me more about 1998 indictment and trial of Susan McDougal.	95.1	What is return of Hong Kong to Chinese sovereignty Kong's population?
95.2	When was Hong Kong return of Hong Kong to Chinese sovereigntyed to Chinese sovereignty?	95.3	Who was the Chinese President at the time of the return of Hong Kong to Chinese sovereignty?
95.4	Regarding return of Hong Kong to Chinese sovereignty, Who was the British Foreign Secretary at the time?	95.5	What other countries formally congratulated China on the return of Hong Kong to Chinese sovereignty?
95.6	Tell me more about return of Hong Kong to Chinese sovereignty.	96.1	What materials was the 1998 Nagano Olympic Games Olympic torch made of?
96.2	How long was the men's downhill ski run in 1998 Nagano Olympic Games?	96.3	Who won gold medals in 1998 Nagano Olympic Games?
96.4	Which country took the first gold medal at 1998 Nagano Olympic Games?	96.5	Regarding 1998 Nagano Olympic Games, Who won the women's giant slalom?
96.6	How many countries were represented at 1998 Nagano Olympic Games?	96.7	Tell me more about 1998 Nagano Olympic Games.
97.1	Who is the lead singer of the Counting Crows?	97.2	What year did Counting Crows form?
97.3	What is the title of Counting Crows signature hit?	97.4	What is the title of the Counting Crows' first record?
97.5	List the Counting Crows' record titles?	97.6	List the Counting Crows' band members?
97.7	Tell me more about Counting Crows.	98.1	When was the American Legion founded?
98.2	Where was the American Legion founded?	98.3	How many members does the American Legion have?
98.4	What organization has helped to revitalize American Legion membership?	98.5	List American Legionnaires?
98.6	Tell me more about American Legion.	99.1	List Woody Guthrie's songs.
99.2	When was Woody Guthrie born?	99.3	Where was Woody Guthrie born?
99.4	What year did Woody Guthrie die?	99.5	Where did Woody Guthrie die?
99.6	What caused Woody Guthrie's death?	99.7	Tell me more about Woody Guthrie.
100.1	Where was Sammy Sosa born?	100.2	What was Sammy Sosa's team?
100.3	How many home runs were hit by Sammy Sosa in 1998?	100.4	Who was Sammy Sosa's competitor for the home run title in 1998?
100.5	Regarding Sammy Sosa, What was the record number of home runs in 1998?	100.6	What award was won by Sammy Sosa in 1998?
100.7	Name the pitchers off of which Sammy Sosa homered?	100.8	Tell me more about Sammy Sosa.
101.1	When was Michael Weiss born?	101.2	Who is Michael Weiss's coach?
101.3	When did Michael Weiss win his first U.S. Skating title?	101.4	When did Michael Weiss win his second U.S. Skating title?
101.5	Who is Michael Weiss's choreographer?	101.6	What is Michael Weiss's home town?
101.7	List Michael Weiss's competitors.	101.8	Tell me more about Michael Weiss.
102.1	What was the official name of the Boston Big Dig Dig?	102.2	When did the Boston Big Dig Dig begin?
102.3	What was the original estimated cost of the Boston Big Dig Dig?	102.4	Regarding Boston Big Dig, What was the expected completion date?
102.5	What is the length of the Boston Big Dig Dig?	102.6	List individuals associated with the Boston Big Dig Dig?
102.7	Tell me more about Boston Big Dig.	103.1	Where was Super Bowl XXXIV held?
103.2	Regarding Super Bowl XXXIV, What team won the game?	103.3	Regarding Super Bowl XXXIV, What was the final score?
103.4	Regarding Super Bowl XXXIV, What was the attendance at the game?	103.5	How many plays were there in Super Bowl XXXIV?
103.6	Regarding Super Bowl XXXIV, List players who scored touchdowns in the game?	103.7	Tell me more about Super Bowl XXXIV.
104.1	In what city was the 1999 North American International Auto Show held?	104.2	What type of vehicle dominated the 1999 North American International Auto Show?
104.3	What auto won the 1999 North American International Auto Show American Car of the Year award at the show?	104.4	List 1999 North American International Auto Show manufacturers in the show?
104.5	How many 1999 North American International Auto Showmakers and suppliers had displays at the show?	104.6	What was the expected attendance at the 1999 North American International Auto Show?
104.7	In what year was the first 1999 North American International Auto Show Show held?	104.8	Tell me more about 1999 North American International Auto Show.
105.1	In what 1980 Mount St. Helens eruptionain range is Mt. St. Helens located?	105.2	Who named 1980 Mount St. Helens eruption St. Helens?

ID		ID	
105.3	What was the date of Mt. 1980 Mount St. Helens eruption. Helens' eruption?	105.4	How many people died when 1980 Mount St. Helens eruption erupted?
105.5	What was the height of the volcano after the 1980 Mount St. Helens eruption?	105.6	List names of eyewitnesses of the 1980 Mount St. Helens eruption?
105.7	Tell me more about 1980 Mount St. Helens eruption.	106.1	Regarding 1998 Baseball World Series, What is the name of the winning team?
106.2	Regarding 1998 Baseball World Series, What is the name of the losing team?	106.3	Regarding 1998 Baseball World Series, Who was named Most Valuable Player (MVP)?
106.4	How many games were played in the 1998 Baseball World Series?	106.5	Regarding 1998 Baseball World Series, What is the name of the winning manager?
106.6	Name the players in the 1998 Baseball World Series?	106.7	Tell me more about 1998 Baseball World Series.
107.1	How long is the Chunnel?	107.2	Regarding Chunnel, What year did construction of the tunnel begin?
107.3	What year did the Chunnel open for traffic?	107.4	How many people use the Chunnel each year?
107.5	Who operates the Chunnel?	107.6	List dates of Chunnel closures.
107.7	Tell me more about Chunnel.	108.1	Who is the parent company of Sony Pictures Entertainment (SPE) Pictures?
108.2	What U.S. company did Sony Pictures Entertainment (SPE) purchase to form SPE?	108.3	Name the president and COO of the Sony Pictures Entertainment (SPE)?
108.4	Name movies released by Sony Pictures Entertainment (SPE)?	108.5	Name TV shows by the Sony Pictures Entertainment (SPE)?
108.6	Who is the vice-president of Sony Pictures Entertainment (SPE)?	108.7	Tell me more about Sony Pictures Entertainment (SPE).
109.1	How many customers does Telefonica of Spain have?	109.2	How many countries does Telefonica of Spain operate in?
109.3	How is Telefonica of Spain ranked in size among the world's telecommunications companies?	109.4	Regarding Telefonica of Spain, Name the chairman?
109.5	Name companies involved in mergers with Telefonica of Spain.	109.6	Tell me more about Telefonica of Spain.
110.1	What is the mission of the Lions Club International Club?	110.2	When was the Lions Club International founded?
110.3	Where is the Lions Club International's world-wide headquarters?	110.4	Who is the Lions Club International Club president?
110.5	Name officials of the Lions Club International?	110.6	Name programs sponsored by the Lions Club International Club?
110.7	Tell me more about Lions Club International.	111.1	When was AMWAY founded?
111.2	Where is AMWAY headquartered?	111.3	Who is the president of AMWAY?
111.4	Name the officials of AMWAY?	111.5	What is the name "AMWAY" short for?
111.6	Tell me more about AMWAY.	112.1	When did the first McDonald's Corporation's restaurant open in the U.S?
112.2	Regarding McDonald's Corporation, Where is the headquarters located?	112.3	What is the McDonald's Corporation's annual revenue?
112.4	Who made McDonald's Corporation's the largest fast-food chain?	112.5	Name the McDonald's Corporation's top officials?
112.6	Name the non-hamburger restaurant holdings of the McDonald's Corporation?	112.7	Tell me more about McDonald's Corporation.
113.1	What is Paul Newman primary career?	113.2	What is Paul Newman second successful career?
113.3	What is the name of Paul Newman that he started?	113.4	Name the camps started under Paul Newman Hole in the Wall Foundation?
113.5	Name some of Paul Newman movies?	113.6	Who is Paul Newman married to?
113.7	Tell me more about Paul Newman.	114.1	What is Jesse Ventura political party affiliation?
114.2	What is Jesse Ventura birth name?	114.3	List Jesse Ventura various occupations?
114.4	Name movies/TV shows Jesse Ventura appeared in?	114.5	What is Jesse Ventura wife's name?
114.6	How many children do Jesse Ventura have?	114.7	Tell me more about Jesse Ventura.
115.1	Regarding Longwood Gardens, When was the initial land purchased?	115.2	Where is Longwood Gardens?
115.3	How large is Longwood Gardens?	115.4	Who created Longwood Gardens?
115.5	How many visitors does Longwood Gardens get per year?	115.6	When is the best month to visit the Longwood Gardens?
115.7	List personnel of the Longwood Gardens?	115.8	Tell me more about Longwood Gardens.
116.1	Where is Camp David?	116.2	How large is Camp David?
116.3	What was Camp David originally called?	116.4	When was Camp David first used?

ID		ID	
116.5	What U.S. President first used Camp David?	116.6	Regarding Camp David, Who are some world leaders that have met there?
116.7	Tell me more about Camp David.	117.1	What kind of plant is kudzu?
117.2	When was kudzu introduced into the U.S?	117.3	From where was kudzu introduced?
117.4	What are other names kudzu is known by?	117.5	Why is kudzu a problem?
117.6	What has been found to kill kudzu?	117.7	Tell me more about kudzu.
118.1	When was U.S. Medal of Honor first awarded?	118.2	Who authorized U.S. Medal of Honor?
118.3	How many have received the award U.S. Medal of Honorince 1863?	118.4	What U.S. Medal of Honor of Honor recipients are in Congress?
118.5	Who is the only woman to receive U.S. Medal of Honor?	118.6	How many veterans have received the U.S. Medal of Honor twice?
118.7	Tell me more about U.S. Medal of Honor.	119.1	When was Harley-Davidson founded?
119.2	Where is Harley-Davidson based?	119.3	They are best known for making what product? Harley-Davidson are best known for making what product?
119.4	What other products do Harley-Davidson produce?	119.5	What is the average age of a Harley-Davidson rider?
119.6	What company did Harley-Davidson buy out?	119.7	Tell me more about Harley-Davidson.
120.1	What was Rose Crumb occupation?	120.2	Where was Rose Crumb from?
120.3	What organization did Rose Crumb found?	120.4	When did Rose Crumb found it?
120.5	What awards has Rose Crumb received?	120.6	How old was Rose Crumb when she won the awards?
120.7	Tell me more about Rose Crumb.	121.1	What was Rachel Carson vocation?
121.2	Where was Rachel Carson home?	121.3	What books did Rachel Carson write?
121.4	When did Rachel Carson write her book exposing dangers of pesticides?	121.5	Her book caused what pesticide to be banned? Rachel Carson book caused what pesticide to be banned?
121.6	What did Rachel Carson die of?	121.7	When did Rachel Carson die?
121.8	Tell me more about Rachel Carson.	122.1	When was Paul Revere born?
122.2	When did Paul Revere die?	122.3	In what cemetery is Paul Revere buried?
122.4	When did Paul Revere make his famous ride?	122.5	From where did Paul Revere begin his famous ride?
122.6	Where did Paul Revere famous ride end?	122.7	What were some of Paul Revere occupations?
122.8	Tell me more about Paul Revere.	123.1	When was Vicente Fox born?
123.2	Where was Vicente Fox educated?	123.3	Of what country is Vicente Fox president?
123.4	What job did Vicente Fox hold before becoming president?	123.5	What countries did Vicente Fox visit after election?
123.6	Tell me more about Vicente Fox.	124.1	When was Rocky Marciano born?
124.2	Where did Rocky Marciano live?	124.3	When did Rocky Marciano die?
124.4	How did Rocky Marciano die?	124.5	How many fights did Rocky Marciano win?
124.6	Who were some of Rocky Marciano opponents?	124.7	Tell me more about Rocky Marciano.
125.1	What operas has Enrico Caruso sung?	125.2	Whom did Enrico Caruso marry?
125.3	How many children did Enrico Caruso have?	125.4	How many opening season performances did Enrico Caruso have at the Met?
125.5	How many performances did Enrico Caruso sing at the Met?	125.6	At what age did Enrico Caruso die?
125.7	Where did Enrico Caruso die?	125.8	Tell me more about Enrico Caruso.
126.1	When was he elected Pope Pius XII?	126.2	What was his name before becoming Pope Pius XII?
126.3	What official positions did he hold prior to becoming Pope Pius XII XII?	126.4	How long was Pope Pius XII pontificate?
126.5	How many people did Pope Pius XII canonize?	126.6	What caused the death of Pope Pius XII XII?
126.7	What Pope Pius XII followed Pius XII?	126.8	Tell me more about Pope Pius XII.
127.1	Where is the U.S. Naval Academy?	127.2	When was U.S. Naval Academy founded?
127.3	Regarding U.S. Naval Academy, What is the enrollment?	127.4	What are the U.S. Naval Academytudents called?
127.5	Who is the father of the U.S. Naval Academy.S. Navy?	127.6	List people who have attended the U.S. Naval Academy?
127.7	Tell me more about U.S. Naval Academy.	128.1	What does OPEC stand for?
128.2	How many countries are members of OPEC?	128.3	What countries constitute the OPEC committee?
128.4	Where is the headquarters of OPEC located?	128.5	List OPEC countries.
128.6	Tell me more about OPEC.	129.1	What does the acronym NATO stand for?

ID		ID	
129.2	When was NATO established?	129.3	Where was the agreement establishing NATO signed?
129.4	Regarding NATO, Which countries were the original signers?	129.5	Where is NATO headquartered?
129.6	Tell me more about NATO.	130.1	What causes tsunamis?
130.2	Where does tsunami commonly occur?	130.3	What is tsunami maximum height?
130.4	How fast can tsunami travel?	130.5	What countries has tsunami struck?
130.6	What language does the term "tsunami" come from?	130.7	Tell me more about tsunami.
131.1	What type of craft was the Hindenburg disaster?	131.2	How fast could Hindenburg disaster travel?
131.3	When did the Hindenburg disaster occur?	131.4	Where did the Hindenburg disaster occur?
131.5	Regarding Hindenburg disaster, How many people were on board?	131.6	How many of Hindenburg disaster were killed?
131.7	Name individuals who witnessed the Hindenburg disaster?	131.8	Tell me more about Hindenburg disaster.
132.1	When was Kim Jong Il born?	132.2	Who is Kim Jong Il's father?
132.3	What country does Kim Jong Il rule?	132.4	What posts has Kim Jong Il held in the government of this country?
132.5	To whom is Kim Jong Il married?	132.6	Tell me more about Kim Jong Il.
133.1	Where did this Hurricane Mitch occur?	133.2	When did this Hurricane Mitch occur?
133.3	As of the time of Hurricane Mitch, what previous hurricanes had higher death totals?	133.4	What countries offered aid for this Hurricane Mitch?
133.5	What country had the highest death total from this Hurricane Mitch?	133.6	Tell me more about Hurricane Mitch.
134.1	What is a genome?	134.2	List species whose genomes have been sequenced.
134.3	List the organizations that sequenced the Human genome.	134.4	How many chromosomes does the Human genome contain?
134.5	What is the length of the Human genome?	134.6	Tell me more about genome.
135.1	What country was the primary beneficiary of this Food-for-Oil Agreement?	135.2	Who authorized this Food-for-Oil Agreement?
135.3	When was this Food-for-Oil Agreement authorized?	135.4	When was this Food-for-Oil Agreement signed?
135.5	What countries participated in this Food-for-Oil Agreement by providing food or medicine?	135.6	Tell me more about Food-for-Oil Agreement.
136.1	Who was the first Imam of the Shiite sect of Islam?	136.2	Where is Shiite tomb?
136.3	Regarding Shiite, What was this person's relationship to the Prophet Mohammad?	136.4	Who was the third Imam of Shiite Muslims?
136.5	When did Shiite die?	136.6	What portion of Muslims are Shiite?
136.7	What Shiite leaders were killed in Pakistan?	136.8	Tell me more about Shiite.
137.1	What is the former name of Kinmen Island?	137.2	What country governs Kinmen Island?
137.3	What other Kinmen Island groups are controlled by this government?	137.4	In the 1950's, who regularly bombarded Kinmen Island?
137.5	How far is Kinmen Island from this country?	137.6	Of the two governments involved over Kinmen Island, which has air superiority?
137.7	Tell me more about Kinmen Island.	138.1	When was the International Bureau of Universal Postal Union (UPU) organized?
138.2	When did the International Bureau of Universal Postal Union (UPU) become part of the UN?	138.3	Where were International Bureau of Universal Postal Union (UPU) congresses held?
138.4	When did China first join the International Bureau of Universal Postal Union (UPU)?	138.5	Who is the Director-General of the International Bureau of Universal Postal Union (UPU)?
138.6	Tell me more about International Bureau of Universal Postal Union (UPU).	139.1	When was the Organization of Islamic Conference (OIC) of Islamic Conference organized?
139.2	Which countries are members of the Organization of Islamic Conference (OIC)?	139.3	Who has served as Secretary General of the Organization of Islamic Conference (OIC)?
139.4	Where was the 8th summit of the Organization of Islamic Conference (OIC) held?	139.5	Where was the 24th Organization of Islamic Conference (OIC) Conference of Foreign Ministers of the OIC held?
139.6	Tell me more about Organization of Islamic Conference (OIC).	140.1	What government organization goes by the acronym PBGC?
140.2	Who is the head of PBGC?	140.3	When was PBGC established?
140.4	Employees of what companies are receiving benefits from PBGC?	140.5	What is the average waiting time for PBGC to determine benefits?

ID		ID	
140.6	Tell me more about PBGC.	141.1	What position did Warren Moon play in professional football?
141.2	Where did Warren Moon play in college?	141.3	In what year was Warren Moon born?
141.4	How many times was Warren Moon a Pro Bowler?	141.5	Who is Warren Moon's agent?
141.6	Who have coached Warren Moon in professional football?	141.7	List the professional teams for which Warren Moon has been a player?
141.8	Tell me more about Warren Moon.	142.1	What does LPGA stand for?
142.2	Where is the LPGA headquartered?	142.3	How many events are part of the LPGA tour?
142.4	When does the LPGA celebrate its 50th anniversary?	142.5	How many people were founders of LPGA?
142.6	Name past and present LPGA commissioners.	142.7	Name tournaments in which LPGA players have participated.
142.8	Tell me more about LPGA.	143.1	Where is the American Enterprise Institute located?
143.2	What is the full title of American Enterprise Institute?	143.3	When was American Enterprise Institute founded?
143.4	Who is the senior vice president of the American Enterprise Institute?	143.5	What is the zip code of the American Enterprise Institute?
143.6	Who have been "scholars" at the American Enterprise Institute?	143.7	Who have been "fellows" at the American Enterprise Institute?
143.8	Tell me more about American Enterprise Institute.	144.1	Where in the U.S. is the 82nd Airborne Division based?
144.2	When was the 82nd Airborne Division formed?	144.3	How many personnel comprise the 82nd Airborne Division?
144.4	What is the 82nd Airborne Division's motto?	144.5	What is the 82nd Airborne Division's official song?
144.6	In what conflicts has the 82nd Airborne Division participated?	144.7	Who have commanded the 82nd Airborne Division?
144.8	Tell me more about 82nd Airborne Division.	145.1	Regarding John William King convicted of murder, How many non-white members of the jury were there?
145.2	Regarding John William King convicted of murder, Who was the foreman for the jury?	145.3	Regarding John William King convicted of murder, Where was the trial held?
145.4	When was John William King convicted of murder convicted?	145.5	Who was the victim of the John William King convicted of murder?
145.6	Regarding John William King convicted of murder, What defense and prosecution attorneys participated in the trial?	145.7	Tell me more about John William King convicted of murder.
146.1	Regarding Pakistani government overthrown in 1999, Who was the leader deposed by the overthrow?	146.2	When did the overthrown Pakistani government overthrown in 1999 come to power?
146.3	Regarding Pakistani government overthrown in 1999, What group accomplished the overthrow?	146.4	Regarding Pakistani government overthrown in 1999, Who was the nominal leader after the overthrow?
146.5	Regarding Pakistani government overthrown in 1999, For what crime was the deposed leader found guilty?	146.6	Regarding Pakistani government overthrown in 1999, Which countries formally disapproved of the overthrow?
146.7	Tell me more about Pakistani government overthrown in 1999.	147.1	When did Britain's Prince Edward marries Edward engage to marry?
147.2	Who did the Britain's Prince Edward marries marry?	147.3	Where did Britain's Prince Edward marries honeymoon?
147.4	Where was Britain's Prince Edward marries in line for the throne at the time of the wedding?	147.5	What was the Britain's Prince Edward marries's occupation?
147.6	Regarding Britain's Prince Edward marries, How many people viewed the wedding on television?	147.7	Regarding Britain's Prince Edward marries, What individuals were at the wedding?
147.8	Tell me more about Britain's Prince Edward marries.	148.1	In what country is tourists massacred at Luxor in 1997?
148.2	Regarding tourists massacred at Luxor in 1997, On what date did the massacre happen?	148.3	Regarding tourists massacred at Luxor in 1997, How many people died from the massacre?
148.4	At what site in tourists massacred at Luxor in 1997 did the massacre occur?	148.5	Regarding tourists massacred at Luxor in 1997, Which terrorist organization claimed responsibility for the massacre?

ID		ID	
148.6	Tourists from which countries were among the dead? tourists massacred at Luxor in 1997 from which countries were among the dead?	148.7	Tell me more about tourists massacred at Luxor in 1997.
149.1	The Daily Show appears on what cable channel?	149.2	The Daily Show parodies what other type of TV program?
149.3	Who is host of The Daily Show?	149.4	At what time is The Daily Show initially televised?
149.5	Who is the creator of The Daily Show?	149.6	What was the title for The Daily Show's 2000 election coverage?
149.7	What celebrities have appeared on The Daily Show?	149.8	Tell me more about The Daily Show.
150.1	Regarding television show Cheers, In what year was the final episode telecast?	150.2	What network aired the television show Cheers?
150.3	Regarding television show Cheers, Who played Diane Chambers?	150.4	Regarding television show Cheers, Who played Sam Malone the bartender?
150.5	What Boston bar was the inspiration for television show Cheers' bar?	150.6	Regarding television show Cheers, What year was the program first broadcast?
150.7	Name supporting actors who performed in television show Cheers?	150.8	Tell me more about television show Cheers.
151.1	How many races are part of the Winston Cup series?	151.2	What races are part of the Winston Cup series?
151.3	Which drivers have won the Winston Cup?	151.4	What tobacco company is a sponsor of the Winston Cup series?
151.5	On what day of the week are races for the Winston Cup run?	151.6	What is considered the minor league for the Winston Cup series?
151.7	Tell me more about Winston Cup.	152.1	Where was Wolfgang Amadeus Mozart born?
152.2	When was Wolfgang Amadeus Mozart born?	152.3	What year did Wolfgang Amadeus Mozart die?
152.4	List Wolfgang Amadeus Mozart's operas?	152.5	Who was Wolfgang Amadeus Mozart's rival?
152.6	How many children did Wolfgang Amadeus Mozart have?	152.7	Tell me more about Wolfgang Amadeus Mozart.
153.1	When was Alfred Hitchcock born?	153.2	Where was Alfred Hitchcock born?
153.3	What was Alfred Hitchcock's first movie?	153.4	List Alfred Hitchcock movie nominations for best director?
153.5	How many Oscar awards did Alfred Hitchcock win?	153.6	When did Alfred Hitchcock die?
153.7	Where did Alfred Hitchcock die?	153.8	Tell me more about Alfred Hitchcock.
154.1	What year was Christopher Reeve paralyzed?	154.2	How many "Superman" movies did Christopher Reeve make?
154.3	Regarding Christopher Reeve, During what years were these "Superman" movies made?	154.4	Which actress co-starred in the most "Superman" movies with Christopher Reeve?
154.5	What year did Christopher Reeve commence his theatrical career?	154.6	List titles of movies, other than "Superman" movies, that Christopher Reeve acted in.
154.7	Tell me more about Christopher Reeve.	155.1	Hugo Chavez is president of which country?
155.2	How old was Hugo Chavez when first elected?	155.3	Where did Hugo Chavez graduate from?
155.4	What year did Hugo Chavez graduate from this school?	155.5	List countries visited by Hugo Chavez?
155.6	What percentage of the 2000 Presidential vote did Hugo Chavez receive?	155.7	Who was Hugo Chavez's opponent in his 1998 Presidential race?
155.8	Tell me more about Hugo Chavez.	156.1	Who founded NASCAR?
156.2	When was NASCAR founded?	156.3	Who took control of NASCAR from the founder?
156.4	List winners of the NASCAR races.	156.5	How much money does NASCAR generate from TV rights annually?
156.6	Who holds the record of career victories in NASCAR?	156.7	What is the record career total of victories in NASCAR?
156.8	Tell me more about NASCAR.	157.1	What was the number of member United Nations (U.N.) of the U.N. in 2000?
157.2	How many United Nations (U.N.)onpermanent members are in the Security Council?	157.3	How long does a United Nations (U.N.)onpermanent member serve on the Security Council?
157.4	Regarding United Nations (U.N.), How often does the presidency of the Security Council rotate?	157.5	Who was the President of the United Nations (U.N.).N. Security Council for August 1999?
157.6	Who has served as Secretary-General of the United Nations (U.N.).N?	157.7	Regarding United Nations (U.N.), Who is the Secretary-General for political affairs?

ID		ID	
157.8	Tell me more about United Nations (U.N.).	158.1	Who became Tufts University President in 1992?
158.2	Over which other Tufts University did he preside?	158.3	What was Tufts University' endowment in 1992 when he became president?
158.4	What is Tufts University' current endowment?	158.5	How many schools make up the Tufts University?
158.6	Name the schools of Tufts University.	158.7	Where is Tufts University located?
158.8	Tell me more about Tufts University.	159.1	What year was Wal-Mart founded?
159.2	Who founded Wal-Mart?	159.3	List companies that have filed suits against Wal-Mart.
159.4	How many Wal-Mart employees are there in the U.S.?	159.5	How many stores does Wal-Mart operate world-wide?
159.6	How many Wal-Mart outlets are there in India?	159.7	How many Wal-Mart outlets are there in China?
159.8	Tell me more about Wal-Mart.	160.1	How many member nations are in the IMF?
160.2	List the countries that have been provided loans by the IMF.	160.3	Which country received the largest loan ever granted by the IMF?
160.4	What was the record loan amount granted by the IMF?	160.5	What does IMF stand for?
160.6	When was the IMF founded?	160.7	Where is the IMF headquartered?
160.8	Tell me more about IMF.	161.1	In what city was the 1999 Baseball All-Star Game All-Star Game held?
161.2	In what city was the 1999 Baseball All-Star Game Game originally scheduled?	161.3	List the official sponsors of the 1999 Baseball All-Star Game?
161.4	What is the name of the ballpark where the 1999 Baseball All-Star Game was played?	161.5	Regarding 1999 Baseball All-Star Game, What is the seating capacity of the ballpark?
161.6	What was the date of the 1999 Baseball All-Star Game All-Star Game?	161.7	Who was the Most Valuable Player (MVP) of the 1999 Baseball All-Star Game?
161.8	Tell me more about 1999 Baseball All-Star Game.	162.1	Myeloma is cancer in what part of the body? Multiple Myeloma is cancer in what part of the body?
162.2	What drug is used to treat Multiple Myeloma?	162.3	What other form of treatment has been used for multiple myeloma?
162.4	How many Americans develop multiple myeloma every year?	162.5	How many deaths are caused by multiple myeloma in the U.S. annually?
162.6	List facilities involved in the treatment of multiple myeloma.	162.7	Tell me more about Multiple Myeloma.
163.1	Where is the Hermitage Museum located?	163.2	Along the banks of what river is the Hermitage Museum located?
163.3	In what year did Catherine the Great begin buying for the Hermitage Museum?	163.4	What is the size of the Hermitage Museum collection?
163.5	Regarding Hermitage Museum, List the artists represented in the collection?	163.6	In what city will there be a satellite museum for the Hermitage Museum?
163.7	Who is the director of the Hermitage Museum?	163.8	Tell me more about Hermitage Museum.
164.1	Where was Judi Dench born?	164.2	What movies did Judi Dench play in?
164.3	Who is Judi Dench married to?	164.4	How many Oscars has Judi Dench been nominated for?
164.5	How many Oscars has Judi Dench won?	164.6	How many Tony Awards has Judi Dench won?
164.7	In what year was Judi Dench born?	164.8	Tell me more about Judi Dench.
165.1	In what year was the Queen Mum's 100th Birthday celebrated?	165.2	What color was the dress that the Queen Mum's 100th Birthdayhe wore at her birthday lunch?
165.3	The the Queen Mum's 100th Birthday Mother received congratulatory greetings from what Heads of State?	165.4	What was the the Queen Mum's 100th Birthday Mother's birth name?
165.5	How many interviews has the Queen Mum's 100th Birthdayhe granted the Press in her 100 years on earth?	165.6	What was her husband's title when the Queen Mum's 100th Birthdayhe married him?
165.7	What was his title when the Queen Mum's 100th Birthday died?	165.8	Tell me more about the Queen Mum's 100th Birthday.
166.1	How many humans were infected with avian flu outbreak in Hong Kong flu in Hong Kong in 1997?	166.2	How many humans died of avian flu outbreak in Hong Kong flu in Hong Kong in 1997?
166.3	Regarding avian flu outbreak in Hong Kong, How may chickens were slaughtered to stop further spread of the disease to humans?	166.4	What strain of avian flu outbreak in Hong Kong flu broke out in Hong Kong in 1997?

ID		ID	
166.5	What new strain of avian flu outbreak in Hong Kong flu appeared in Hong Kong within the next two years?	166.6	What vaccines are known to be effective against avian flu outbreak in Hong Kong flu?
166.7	Tell me more about avian flu outbreak in Hong Kong.	167.1	In what city is the Millennium Wheel located?
167.2	How high is the Millennium Wheel?	167.3	What is the Millennium Wheel popular name?
167.4	Who built the Millennium Wheel?	167.5	How many passenger capsules does the Millennium Wheel contain?
167.6	List the names of other the Millennium Wheel structures in England?	167.7	Tell me more about the Millennium Wheel.
168.1	Approximately how many paintings has Prince Charles' paintings Charles completed?	168.2	What charities have benefited from the sale or auction of his Prince Charles' paintings?
168.3	What is the record amount ever paid for one of his Prince Charles' paintings?	168.4	What is Prince Charles' paintings usual painting medium?
168.5	What are Prince Charles' paintings usual subjects?	168.6	What other Prince Charles' paintings showed his paintings in a two-prince exhibition with Prince Charles in London?
168.7	Tell me more about Prince Charles' paintings.	169.1	When did the construction of stone circles begin in the UK?
169.2	Approximately how many stone circles have been found in the UK?	169.3	When was stone circleshenge built?
169.4	In what county was stone circleshenge built?	169.5	What are the locations or names of other stone circles in the UK?
169.6	What is the oldest stone circles circle in the UK?	169.7	Tell me more about stone circles.
170.1	Regarding John Prine, How many songs are on the album "In Spite of Ourselves"?	170.2	What John Prine song was a #1 hit for George Strait?
170.3	In what year did John Prine have a cancerous tumor removed?	170.4	For how many years did John Prine smoke?
170.5	What artists has John Prine done duets with?	170.6	What are the titles of songs written by John Prine?
170.7	Tell me more about John Prine.	171.1	When was Stephen Wynn born?
171.2	What company acquired Stephen Wynn's Mirage Resorts in 2000?	171.3	What is the name of the new Stephen Wynn casino opened in Las Vegas in 1998?
171.4	Who was the talent manager hired by Stephen Wynn to start new theaters, arenas and cabarets?	171.5	What is the name of Stephen Wynn's newest casino in Mississippi?
171.6	Name famous artists whose works have been purchased by Stephen Wynn or are displayed in his galleries.	171.7	Tell me more about Stephen Wynn.
172.1	In what city and Ben & Jerry'state did the business begin?	172.2	What do Ben & Jerry's and Jerry produce?
172.3	What is Ben & Jerry's's last name?	172.4	What is Ben & Jerry's's last name?
172.5	What rock band had a Ben & Jerry's flavor named after them?	172.6	Unilever purchased Ben & Jerry's in 2000 for what price?
172.7	Name unusual flavors created by Ben & Jerry's.	172.8	Tell me more about Ben & Jerry's.
173.1	In what country is the World Tourism Organization (WTO) headquartered?	173.2	What country was the most popular tourist destination in 1995 according to World Tourism Organization (WTO) reports?
173.3	Who was the director of the World Tourism Organization (WTO) in 1996?	173.4	How many tourists visited China in 1995 according to World Tourism Organization (WTO) reports?
173.5	What continent had the highest World Tourism Organization (WTO) growth rate in 1997 according to WTO reports?	173.6	The World Tourism Organization (WTO) has held meetings in what countries?
173.7	Who has served as secretary general of the World Tourism Organization (WTO)?	173.8	Tell me more about World Tourism Organization (WTO).
174.1	When was the American Farm Bureau Federation (AFBF) founded?	174.2	Who was president of the American Farm Bureau Federation (AFBF) in 1999?
174.3	The American Farm Bureau Federation (AFBF) represents how many families?	174.4	What was the total amount in disaster aid the American Farm Bureau Federation (AFBF) asked congress for in July 1999?
174.5	According to the American Farm Bureau Federation (AFBF), what is the average cost of a turkey dinner with all the trimmings?	174.6	What employees of the American Farm Bureau Federation (AFBF) have been mentioned in the news?
174.7	Tell me more about American Farm Bureau Federation (AFBF).	175.1	How old was repatriation of Elian Gonzales at the time of the shipwreck?

ID		ID	
175.2	Who was repatriation of Elian Gonzales staying with in Florida?	175.3	Where was repatriation of Elian Gonzales's father at the time?
175.4	Regarding repatriation of Elian Gonzales, Who was the U.S. Attorney General at the time?	175.5	On what date did repatriation of Elian Gonzales return to Cuba?
175.6	What are the names of repatriation of Elian Gonzales's relatives?	175.7	Tell me more about repatriation of Elian Gonzales.
176.1	Name cast members of the movie "An Officer and a Gentleman".	176.2	Regarding An Officer and a Gentleman, Who directed the movie?
176.3	Regarding An Officer and a Gentleman, In what year did the movie win academy awards?	176.4	Regarding An Officer and a Gentleman, What song from the movie won an academy award?
176.5	Regarding An Officer and a Gentleman, Who sings the song during the final credits?	176.6	Regarding An Officer and a Gentleman, What Florida bar was featured in the movie?
176.7	Tell me more about An Officer and a Gentleman.	177.1	What game can "Deep Blue" play?
177.2	Who won the match between Gary Kasparov and Deep Blue in February 1996?	177.3	Regarding Deep Blue, How much money was wagered in the 1996 match?
177.4	Regarding Deep Blue, The 1996 match consisted of how many games?	177.5	Who won the match between Gary Kasparov and Deep Blue in May 1997?
177.6	How many "moves per second" could Deep Blue scan in 1997?	177.7	In what cities were the matches between Deep Blue and Kasparov held?
177.8	Tell me more about Deep Blue.	178.1	How many methamphetamine labs were dismantled in Arizona in 1998?
178.2	How many methamphetamine labs were dismantled nationwide in the U.S. in 1998?	178.3	How much money did the Clinton Administration budget in 2000 for shutting down methamphetamine labs?
178.4	In what Texas town did the FBI and ATF destroy a compound killing dozens, but found no methamphetamine labs lab?	178.5	What chemical company's labels were found on bottles in a methamphetamine labs lab?
178.6	In what cities or towns have illegal methamphetamine labs been found?	178.7	Tell me more about methamphetamine labs.
179.1	What was the date of Hedy Lamarr's birth?	179.2	What was Hedy Lamarr name at birth?
179.3	In what city was Hedy Lamarr born?	179.4	On what date did Hedy Lamarr die?
179.5	Name movies Hedy Lamarr appeared in.	179.6	What did Hedy Lamarr invent?
179.7	Name Hedy Lamarr's husbands.	179.8	Tell me more about Hedy Lamarr.
180.1	In what city is the Lebanese Parliament located?	180.2	How many seats are in the Lebanese Parliament?
180.3	Who is the Speaker of the Lebanese Parliament?	180.4	Who is the Deputy Speaker of the Lebanese Parliament?
180.5	Name members of the Lebanese Parliament.	180.6	How many years are in a term of the Lebanese Parliament?
180.7	Tell me more about Lebanese Parliament.	181.1	What British league does Manchester United Football Club United play in?
181.2	What is the name of the stadium in Manchester United Football Club where Manchester United plays?	181.3	Who is the manager of Manchester United Football Club United?
181.4	Who is the captain of Manchester United Football Club United?	181.5	Who is the keeper for Manchester United Football Club United?
181.6	Who were leading players for Manchester United Football Club United in the 1990's?	181.7	What is the feeder club for Manchester United Football Club United?
181.8	Which British teams has Manchester United Football Club United played?	181.9	Tell me more about Manchester United Football Club.
182.1	How many performers appeared at the 1998 Edinburgh Fringe?	182.2	What was the total number of performances at the 1998 Edinburgh Fringe?
182.3	What plays were performed at the 1998 Edinburgh Fringe?	182.4	In what month is the 1998 Edinburgh Fringe Fringe held?
182.5	In what country is the 1998 Edinburgh Fringe Fringe held?	182.6	In what year was the 1998 Edinburgh Fringe Fringe begun?
182.7	Tell me more about 1998 Edinburgh Fringe.	183.1	On what date was Thabo Mbeki elected president of South Africa?
183.2	Whom did Thabo Mbeki elected president of South Africa Mbeki succeed as president of South Africa?	183.3	What political party does Thabo Mbeki elected president of South Africa Mbeki belong to?
183.4	What national leaders and spokespersons sent congratulatory messages following Thabo Mbeki elected president of South Africa Mbeki's election as president of South Africa?	183.5	What was the biggest black opposition party at the time of Thabo Mbeki elected president of South Africa's election?

ID		ID	
183.6	How old was Thabo Mbeki elected president of South Africa Mbeki when he was elected president?	183.7	How many seats did Thabo Mbeki elected president of South Africa's party win in the election?
183.8	Tell me more about Thabo Mbeki elected president of South Africa.	184.1	What was the date of the 1999 Chicago Marathon?
184.2	What was the distance of the 1999 Chicago Marathon?	184.3	Who was the men's winner of the 1999 Chicago Marathon?
184.4	Who was the women's winner of the 1999 Chicago Marathon?	184.5	What was the men's winning time in the 1999 Chicago Marathon?
184.6	What was the women's winning time in the 1999 Chicago Marathon?	184.7	Name nations represented in the 1999 Chicago Marathon.
184.8	Tell me more about 1999 Chicago Marathon.	185.1	In what city does the Iditarod Race start?
185.2	In what city does the Iditarod Race end?	185.3	In what month is Iditarod Race held?
185.4	Who is the founder of the Iditarod Race?	185.5	Name people who have won the Iditarod Race?
185.6	How many miles long is the Iditarod Race?	185.7	What is the record time in which the Iditarod Race was won?
185.8	Which companies have sponsored the Iditarod Race?	185.9	Tell me more about Iditarod Race.
186.1	How many Pyramids of Egypt have been discovered in Egypt?	186.2	In what city are the three Great Pyramids of Egypt located?
186.3	What are the names of the three Great Pyramids of Egypt?	186.4	Regarding Pyramids of Egypt, Which is the largest pyramid ever built?
186.5	Regarding Pyramids of Egypt, What is the height of the largest pyramid?	186.6	How old are the three Great Pyramids of Egypt?
186.7	Name additional pyramids of Egypt.	186.8	Tell me more about Pyramids of Egypt.
187.1	In what country is the origin of the Amazon River?	187.2	In what country is the mouth of the Amazon River?
187.3	How long is the Amazon River?	187.4	Name tributaries of the Amazon River.
187.5	In what mountain range does the Amazon River rise?	187.6	What is the name of the Amazon River at its origin?
187.7	Tell me more about Amazon River.	188.1	Regarding avocados, What U.S. state is the highest avocado producer?
188.2	Regarding avocados, What is the fat content of an avocado?	188.3	What are the main commercial varieties of avocados?
188.4	What countries produce avocados?	188.5	Regarding avocados, When was the first avocado plant cultivated in the U.S?
188.6	Regarding avocados, What insect pest threatens avocado crops?	188.7	Tell me more about avocados.
189.1	Regarding Joanne Kathleen Rowling, What are the names of this author's books?	189.2	Who publishes Joanne Kathleen Rowling books?
189.3	How much did this publisher pay for Joanne Kathleen Rowling first manuscript?	189.4	In what year did Joanne Kathleen Rowling start writing her first book?
189.5	For which novel did Joanne Kathleen Rowling receive the Whitbread's Children's Book of the Year award in 2000?	189.6	In what city was Joanne Kathleen Rowling born?
189.7	In what city did Joanne Kathleen Rowling live in 2000?	189.8	Tell me more about Joanne Kathleen Rowling.
190.1	In what city is the company H. J. Heinz Co.eadquartered?	190.2	Who became CEO of the H. J. Heinz Co.mpany in 1998?
190.3	Which H. J. Heinz Co.eir to the company fortune was killed in an aviation accident?	190.4	What year was H. J. Heinz Co.einz Ketchup introduced?
190.5	Which disease does H. J. Heinz Co.einz tout its ketchup as helping to prevent?	190.6	What food companies H. J. Heinz Co.ave been acquired by Heinz?
190.7	Tell me more about H. J. Heinz Co..	191.1	What abbreviation is the International Rowing Federation also known by?
191.2	When was the International Rowing Federation Rowing World Cup first held?	191.3	What country won the first four overall International Rowing Federation Rowing World Cup titles?
191.4	In what cities were International Rowing Federation Rowing World Cup events held?	191.5	Who was elected president of the International Rowing Federation in 1989?
191.6	Regarding International Rowing Federation, How many boats raced in the 2000 Olympic Games in Sydney Australia?	191.7	Tell me more about International Rowing Federation.
192.1	What does the acronym Basque ETA stand for?	192.2	Approximately how many people has Basque ETA killed?
192.3	What are some other Basque ETA separatist groups?	192.4	On what date did Basque ETA announce a cease-fire in 1998?
192.5	Regarding Basque ETA, What date will this cease-fire begin?	192.6	What date did Basque ETA announce that this cease-fire would end?

ID		ID	
192.7	Tell me more about Basque ETA.	193.1	In what city is the World Food Program (WFP) headquartered?
193.2	When was the World Food Program (WFP) created?	193.3	Who was the first director of the World Food Program (WFP)?
193.4	Who was the director of the World Food Program (WFP) in 1997?	193.5	The World Food Program (WFP) is an agency of what organization?
193.6	In 2000, in how many countries did the WFP distribute World Food Program (WFP)?	193.7	What countries have donated to the World Food Program (WFP)?
193.8	Tell me more about World Food Program (WFP).	194.1	Where did the 1996 World Chess Super Tournament take place?
194.2	When did this 1996 World Chess Super Tournament conclude?	194.3	Regarding 1996 World Chess Super Tournament, How many players participated?
194.4	Who did Kasparov defeat in this 1996 World Chess Super Tournament?	194.5	The purpose of this tournament was to help unify what 1996 World Chess Super Tournament chess organizations?
194.6	Tell me more about 1996 World Chess Super Tournament.	195.1	What was the date of the East Timor Independence Timor election on independence?
195.2	What percentage of the vote was for East Timor Independence?	195.3	From what country did East Timor Independence Timor separate?
195.4	In what year did this country invade East Timor Independence Timor?	195.5	On what date did the International Force in East Timor Independence Timor (INTERFET) enter East Timor?
195.6	Regarding East Timor Independence, What countries contributed troops to INTERFET?	195.7	On what date did INTERFET turn over control to UNTAET (U.N. Transitional Authority in East Timor Independence Timor)?
195.8	Tell me more about East Timor Independence.	196.1	On what date was the Adoption of the Euro adopted?
196.2	How many Adoption of the European Union countries originally adopted the Euro?	196.3	Which Adoption of the European Union countries originally chose not to adopt the Euro?
196.4	On what date was Greece allowed to adopt the Adoption of the Euro?	196.5	In what year is the Adoption of the Euro scheduled to completely replace the national currencies?
196.6	In the Danish referendum of 2000, what percentage of the voters rejected adopting the Adoption of the Euro?	196.7	Tell me more about Adoption of the Euro.
197.1	What animal was the first mammal successfully cloned from cloning of mammals (from adult cells) cells?	197.2	Regarding cloning of mammals (from adult cells), What year was this animal born?
197.3	Regarding cloning of mammals (from adult cells), At what institute was this procedure done?	197.4	What other cloning of mammals (from adult cells) have been cloned from adult cells?
197.5	What countries have placed restrictions on human cloning of mammals (from adult cells) research?	197.6	Tell me more about cloning of mammals (from adult cells).
198.1	In what country is this Bushehr Nuclear Facility located?	198.2	What country is assisting with the development of this Bushehr Nuclear Facility?
198.3	What other countries have signed contracts to work on this Bushehr Nuclear Facility?	198.4	What is the claimed primary purpose of this Bushehr Nuclear Facility?
198.5	What model reactor is being planned for this Bushehr Nuclear Facility?	198.6	What year did work on this Bushehr Nuclear Facility originally start?
198.7	Tell me more about Bushehr Nuclear Facility.	199.1	In what year was Padre Pio born?
199.2	Where was Padre Pio born?	199.3	What year did Padre Pio die?
199.4	How old was Padre Pio when he died?	199.5	In what city is Padre Pio's tomb?
199.6	What year did Padre Pio first receive the stigmata?	199.7	Regarding Padre Pio, List other saints who have had the stigmata?
199.8	Tell me more about Padre Pio.	200.1	In what city was Frank Sinatra born?
200.2	What year was Frank Sinatra's first performance?	200.3	What was Frank Sinatra's first recording?
200.4	When did Frank Sinatra die?	200.5	How old was Frank Sinatra when he died?
200.6	Name Frank Sinatra children?	200.7	Tell me more about Frank Sinatra.
201.1	When was William Shakespeare born?	201.2	Where was William Shakespeare born?
201.3	When did William Shakespeare die?	201.4	How old was William Shakespeare when he died?
201.5	How many sonnets did William Shakespeare write?	201.6	How many plays did William Shakespeare write?
201.7	What plays did William Shakespeare write?	201.8	Tell me more about William Shakespeare.

ID		ID	
202.1	Where was Cole Porter born?	202.2	What year was Cole Porter born?
202.3	What year did Cole Porter die?	202.4	What musicals did Cole Porter compose?
202.5	What was Cole Porter's last Broadway show?	202.6	How many songs did Cole Porter write?
202.7	What was Cole Porter's first professional musical?	202.8	Tell me more about Cole Porter.
203.1	What position is Nissan Corp. among Japanese automakers?	203.2	Who is president of Nissan Corp.?
203.3	In what countries does Nissan Corp. manufacture vehicles outside of Japan?	203.4	In what city is Nissan Corp. headquartered?
203.5	What is the Nissan Corp. luxury car?	203.6	What was Nissan Corp. formerly known as?
203.7	Tell me more about Nissan Corp..	204.1	Who founded the Church of Jesus Christ of Latter-day Saints (Mormons) of Jesus Christ of Latter-day Saints?
204.2	In what year was the Church of Jesus Christ of Latter-day Saints (Mormons) founded?	204.3	Where did the Mormon Church of Jesus Christ of Latter-day Saints (Mormons) begin?
204.4	How many members does Church of Jesus Christ of Latter-day Saints (Mormons) have?	204.5	Regarding Church of Jesus Christ of Latter-day Saints (Mormons), Name elected government officials who are Mormon?
204.6	Where is the capital of the Mormon Church of Jesus Christ of Latter-day Saints (Mormons)?	204.7	Regarding Church of Jesus Christ of Latter-day Saints (Mormons), How many temples are in operation worldwide?
204.8	Tell me more about Church of Jesus Christ of Latter-day Saints (Mormons).	205.1	On what island is Mt. 1991 eruption of Mount Pinatubo?
205.2	In what month did 1991 eruption of Mount Pinatubo erupt?	205.3	Regarding 1991 eruption of Mount Pinatubo, How many people were killed?
205.4	Regarding 1991 eruption of Mount Pinatubo, How many people were evacuated?	205.5	How long had Mt. 1991 eruption of Mount Pinatubo been dormant?
205.6	Regarding 1991 eruption of Mount Pinatubo, What other volcanoes are in the Philippines?	205.7	Tell me more about 1991 eruption of Mount Pinatubo.
206.1	When did the Johnstown flood occur?	206.2	How many people perished in the Johnstown flood?
206.3	How much water fell on Johnstown flood?	206.4	What dam caused the Johnstown flood?
206.5	Regarding Johnstown flood, How old was the dam?	206.6	In what state is Johnstown flood?
206.7	Regarding Johnstown flood, What other U.S. states have had dam failures?	206.8	Tell me more about Johnstown flood.
207.1	The Leaning Tower of Pisa of Pisa began to be built in what year?	207.2	How long did it take to build the Leaning Tower of Pisa of Pisa?
207.3	What is the tilt angle of the Leaning Tower of Pisa of Pisa?	207.4	How much was the Leaning Tower of Pisa of Pisa straightened in the 1990's?
207.5	What was the cost of preservation of the Leaning Tower of Pisa of Pisa?	207.6	How high is the Leaning Tower of Pisa of Pisa?
207.7	Name other Leaning Tower of Pisa towers?	207.8	Tell me more about Leaning Tower of Pisa.
208.1	How old is the Great Wall of China?	208.2	How long is the Great Wall of China?
208.3	How long was Great Wall of China used as a defense?	208.4	Under which dynasty was the Great Wall of China built?
208.5	In what cities has the Great Wall of China been found?	208.6	Tell me more about Great Wall of China.
209.1	Where did Carolyn Bessette Kennedy grow up?	209.2	Which high school did Carolyn Bessette Kennedy go to?
209.3	Which college did Carolyn Bessette Kennedy go to?	209.4	Who was Carolyn Bessette Kennedy married to?
209.5	When was Carolyn Bessette Kennedy married?	209.6	On what date did Carolyn Bessette Kennedy die?
209.7	Who were Carolyn Bessette Kennedy family members?	209.8	Tell me more about Carolyn Bessette Kennedy.
210.1	What government position did Janet Reno assume in 1993?	210.2	What position did Janet Reno have immediately prior to 1993?
210.3	Where was Janet Reno father born?	210.4	Where did Janet Reno grow up?
210.5	Who was Janet Reno top Deputy Attorney in 1999?	210.6	What disease was Janet Reno diagnosed with in 1995?
210.7	List universities that Janet Reno visited?	210.8	Tell me more about Janet Reno.
211.1	What year was Patsy Cline inducted into the Hollywood Walk of Fame?	211.2	Who was Patsy Cline married to?
211.3	What year did Patsy Cline die?	211.4	How did Patsy Cline die?

ID		ID	
211.5	What movie was based on Patsy Cline life?	211.6	What was the name of Patsy Cline producer?
211.7	What songs did Patsy Cline record?	211.8	Tell me more about Patsy Cline.
212.1	What year was Barry Manilow born?	212.2	How many times has Barry Manilow married?
212.3	What is the name of the musical that Barry Manilow wrote about the Harmonistas?	212.4	What music school did Barry Manilow attend?
212.5	For what female singer was Barry Manilow the musical director and pianist in the 70's?	212.6	What record label did Barry Manilow sing for in 2000?
212.7	List the songs Barry Manilow recorded?	212.8	Tell me more about Barry Manilow.
213.1	What was Meg Ryan's name at birth?	213.2	Who was Meg Ryan married to?
213.3	When did Meg Ryan meet her husband?	213.4	What movies did Meg Ryan star in?
213.5	In what movie did Meg Ryan own a bookstore?	213.6	Who was Meg Ryan male lead in that movie?
213.7	In what movie did Meg Ryan own a party-planning business?	213.8	Tell me more about Meg Ryan.
214.1	Where was the 2000 Miss America Pageant held?	214.2	Which TV network aired the 2000 Miss America Pageant?
214.3	What was the date of the 2000 Miss America Pageant?	214.4	Who was crowned Miss America 2000 Miss America Pageant?
214.5	What state was Miss America 2000 Miss America Pageant from?	214.6	How many judges were in the 2000 Miss America Pageant?
214.7	Who were the five finalists in the 2000 Miss America Pageant?	214.8	Tell me more about 2000 Miss America Pageant.
215.1	Where was the 1999 Sundance Film Festival held?	215.2	What is the name of the artistic director of the 1999 Sundance Film Festival?
215.3	When was the 1999 Sundance Film Festival held?	215.4	Which actress appeared in two 1999 Sundance Film Festivals shown at the festival?
215.5	Which 1999 Sundance Film Festival won the Dramatic Screen Play Award at the festival?	215.6	Which 1999 Sundance Film Festival won three awards at the festival?
215.7	List films shown at the 1999 Sundance Film Festival.	215.8	Tell me more about 1999 Sundance Film Festival.

Bibliography

[1] Steven Abney, Michael Collins, and Amit Singhal. Answer extraction. In *Association for Computational Linguistics. 6th Applied Natural LanguageProcessing Conference: Proceedings of the Conference*, pages 296–301, Seattle, Washington, USA, 2000.

[2] Michaela Atterer and Hinrich Schütze. An Inverted Index for Storing and Retrieving Grammatical Dependencies. In European Language Resources Association (ELRA), editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may 2008.

[3] Olga Babko-Malaya. Propbank Annotation Guidelines, October 2006.

[4] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[5] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet Project. In *Proceedings of the 17th international conference on Computational Linguistics*, pages 86–90, Morristown, NJ, USA, 1998. Association for Computational Linguistics.

[6] Adam Berger and John Lafferty. Information Retrieval as Statistical Translation. In *Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 222–229, 1999.

[7] Matthew Bilotti and Eric Nyberg. Improving Text Retrieval Precision and Answer Accuracy in Question Answering Systems. In *Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question Answering*, pages 1–8, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

[8] Matthew Bilotti, Le Zhao, Jamie Callan, and Eric Nyberg. Focused Retrieval over Richly-Annotated Collections. In *Proceeding of the SIGIR*

2008 Workshop on Focused Retrieval (Question Answering, Passage Retrieval, Element Retrieval), 2008.

[9] Matthew W. Bilotti, Boris Katz, and Jimmy Lin. What Works Better for Question Answering: Stemming or MorphologicalQuery Expansion? In *IR4QA: Information Retrieval for Question Answering Workshop. SIGIR'04*, 2004.

[10] Matthew W. Bilotti, Paul Ogilvie, Jamie Callan, and Eric Nyberg. Structured retrieval for question answering. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 351–358, New York, NY, USA, 2007. ACM.

[11] Johan Bos, James R. Curran, and Edoardo Guzzetti. The Pronto QA System at TREC 2007: Harvesting Hyponyms, Using Nominalisation Patterns, and Computing Answer Cardinality. In Voorhees et al., editor, *The Sixteenth Text RETrieval Conference, TREC 2007*, 2007.

[12] Gosse Bouma, Ismail Fahmi, Jori Mur, Gertjan van Noord, Lonneke van der Plas, , and Jörg Tiedemann. Linguistic Knowledge and Question Answering. *Traitement Automatique des Langues (TAL)*, 3, 2005.

[13] Gosse Bouma, Gertjan Van Noord, and Robert Malouf. Alpino: Wide-coverage Computational Analysis of Dutch. *Computational Linguistics in the Netherlands 2000. Language and Computers*, 37:45–59(15), 1 December 2001.

[14] Martin Braschler. CLEF 2000 - Overview of Results. In *CLEF '00: Revised Papers from the Workshop of Cross-Language Evaluation Forum on Cross-Language Information Retrieval and Evaluation*, pages 89–101, London, UK, 2001. Springer-Verlag.

[15] Eric Brill. A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92, Third Conference on Applied Natural Language Processing*, pages 152–155, Trento, Italy, 1992.

[16] Eric Brill, Jimmy Lin, Michele Banko, Susan Dumais, and Andrew Ng. Data-Intensive Question Answering. In Ellen M. Voorhees and Donna K. Harman, editors, *Proceedings of TREC-2001, 10th Text Retrieval Conference*, pages 393–400, Gaithersburg, US, 2001. National Institute of Standards and Technology, Gaithersburg, US.

[17] Sabine Buchholz. Using Grammatical Relations, Answer Frequencies and the World Wide Web for TREC Question Answering. In Ellen M. Voorhees and Donna K. Harman, editors, *Proceedings of TREC-2001, 10th Text Retrieval Conference*, pages 502–509, Gaithersburg, US, 2001. National Institute of Standards and Technology, Gaithersburg, US.

[18] R. Burke, K. Hammond, V. Kulyukin, S. Lytinen, N. Tomuro, and S. Schoenberg. Question Answering from Frequently Asked Question Files: Experiences with the FAQ FINDER System. *AI Magazine*, 18(2):57–66, 1997.

[19] Stefan Büttcher, Charles L. A. Clarke, and Ian Soboroff. The TREC 2006 terabyte track. In *Proceedings of the Fifteenth Text REtrieval Conference (TREC 2006)*, 2006.

[20] Jamie Callan, W. Bruce Croft, and Stephen Harding. The INQUERY Retrieval System. In *Proceedings of the 3rd International Conference on Database and Expert Systems Application*, pages 78–83, 1992.

[21] D Carmel, Y S Maarek, Y Mass, N Efraty, and G M Landau. An Extension of the Vector Space Model for Querying XML Documents via XML Fragments. In *Proceedings SIGIR 2002 Workshop on XML and Information Retrieval*, pages 14–25, 2002.

[22] David Carmel, Yoelle S. Maarek, Matan Mandelbrod, Yosi Mass, and Aya Soffer. Searching XML documents via XML fragments. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR con-*

ference on Research and development in informaion retrieval, pages 151–158, New York, NY, USA, 2003. ACM.

[23] Y. Chang, H. Xu, and S. Bai. A re-examination of IR techniques in QA system. In *Natural Language Processing-IJCNLP 2004: First International Joint Conference, Hainan Island, China, March 22-24, 2004, Revised Selected Papers*. Springer, 2005.

[24] J. Chen, A.R. Diekema, M.D. Taffet, N. McCracken, N. Ercan Ozgencil, O. Yilmazel, and E.D. Liddy. Question Answering: CNLP at the TREC-10 Question Answering Track. In Ellen M. Voorhees and Donna K. Harman, editors, *Proceedings of TREC-2001, 10th Text Retrieval Conference*, pages 485–494, Gaithersburg, US, 2001. National Institute of Standards and Technology, Gaithersburg, US.

[25] Peter A. Chew, Brett W. Bader, and Ahmed Abdelali. Latent Morpho-Semantic Analysis: Multilingual Information Retrieval with Character N-Grams and Mutual Information. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 129–136, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

[26] Charles L. A. Clarke and Egidio 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 informaion retrieval*, pages 427–428, New York, NY, USA, 2003. ACM.

[27] J. Clément, P. Flajolet, and B. Vallée. The Analysis of Hybrid Trie Structures. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 531–539, Philadelphia:PA, USA, 1998. SIAM Press.

[28] Hang Cui, Keya Li, Renxu Sun, Tat-Seng Chua, and Min-Yen Kan.

National University of Singapore at the TREC-13 Question Answering Main Task. In *Text REtrieval Conference (TREC 2004)*, 2004.

[29] Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. Question answering passage retrieval using dependency relations. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 400–407, New York, NY, USA, 2005. ACM.

[30] Hoa Trang Dang, Diane Kelly, and Jimmy Lin. Overview of the TREC 2007 Question Answering Track. In *Proceedings of the Text REtrieval Conference*, 2007.

[31] Hoa Trang Dang and Jimmy Lin. Different Structures for Evaluating Answers to Complex Questions: Pyramids Won't Topple, and Neither Will Human Assessors. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 768–775, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[32] Hoa Trang Dang, Jimmy Lin, and Diane Kelly. Overview of the TREC 2006 Question Answering Track. In *Proceedings of the Text REtrieval Conference*, 2006.

[33] César de Pablo-Sánchez, Ana González-Ledesma, Antonio Moreno-Sandoval, and Maria Teresa Vicente-Díez. MIRACLE Experiments in QA@ CLEF 2006 in Spanish: Main Task, Real-Time QA and Exploratory QA Using Wikipedia (WiQA). In *Evaluation of Multilingual and Multi-modal Information Retrieval 7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006, Alicante, Spain, September 20-22, 2006: Revised Selected Papers*, pages 463–472. Springer, 2007.

[34] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[35] Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin, and Andrew Ng. Web question answering: is more always better? In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 291–298. ACM Press, 2002.

[36] David Elworthy. Answering using a large NLP System. In Ellen M. Voorhees and Donna K. Harman, editors, *Proceedings of TREC-9, 9th Text Retrieval Conference*, pages 355–360, Gaithersburg, US, 2000. National Institute of Standards and Technology, Gaithersburg, US.

[37] Chris Faloutsos and Stavros Christodoulakis. Signature files: an access method for documents and its analytical performance evaluation. *ACM Transactions on Information Systems (TOIS)*, 2(4):267–288, 1984.

[38] Christiane Fellbaum. *WordNet – An electronic lexical database*. MIT Press, Cambridge, Massachusetts and London, England, 1998.

[39] Charles J. Fillmore. Frame semantics and the nature of language. In *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, volume 280, pages 20–32, 1976.

[40] Michael Fleischman, Eduard Hovy, and Abdessamad Echihabi. Offline Strategies for Online Question Answering: Answering Questions Before They Are Asked. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 1–7, Sapporo Convention Center, Sapporo, Japan, 2003.

[41] Pamela Forner, Anselmo Peñas, Iñaki Alegria, Corina Forăscu, Nicolas Moreau, Petya Osenova, Prokopis Prokopidis, Paulo Rocha, Bogdan Sacaleanu, Richard Sutcliffe, and Erik Tjong Kim Sang. Overview of the CLEF 2008 Multilingual Question Answering Track. In *Working Notes for the CLEF 2008 Workshop, 17-19 September, Aarhus, Denmark*, 2008.

[42] N. Fuhr. Models in information retrieval. In M. Agosti, F. Crestani, and G Pasi, editors, *Lectures in Information Retrieval*, pages 21–50. Springer, Heidelberg et al., 2001.

[43] Norbert Fuhr and Norbert Gövert. Index compression vs. retrieval time of inverted files for XML documents. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 662–664, New York, NY, USA, 2002. ACM.

[44] Norbert Fuhr and Norbert Gövert. Index Compression vs. Retrieval Time of Inverted Files for XML Documents. Technical report, University of Dortmund, 2002.

[45] Norbert Fuhr, Norbert Gövert, Gabriella Kazai, and Mounia Lalmas, editors. *Proceedings of the First Workshop of the INitiative for the Evaluation of XML Retrieval (INEX), Schloss Dagstuhl, Germany, December 9-11, 2002*, 2002.

[46] Junichi Fukumoto, Tsuneaki Kato, and Fumito Masui. An evaluation of question answering challenge (QAC-1) at the NTCIR workshop 3. *SIGIR Forum*, 38(1):25–28, 2004.

[47] Danilo Giampiccolo, Pamela Forner, Jesús Herrera, Anselmo Peñas, Christelle Ayache, Corina Forascu, Valentin Jijkoun, Petya Osenova, Paulo Rocha, Bogdan Sacaleanu, and Richard F. E. Sutcliffe. Overview of the CLEF 2007 Multilingual Question Answering Track. In Carol Peters, Valentin Jijkoun, Thomas Mandl, Henning Müller, Douglas W. Oard, Anselmo Peñas, Vivien Petras, and Diana Santos, editors, *CLEF*, volume 5152 of *Lecture Notes in Computer Science*, pages 200–236. Springer, 2007.

[48] Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, 2002.

[49] Norbert Gövert and Gabriella Kazai. Overview of the Initiative for the

Evaluation of XML retrieval (INEX) 2002. In Fuhr et al. [45], pages 1–17.

[50] David Graff. The AQUAINT Corpus of English News Text. CDROM, 2002. ISBN: 1-58563-240-6.

[51] Mark A. Greenwood. Using Pertainyms to Improve Passage Retrieval for Questions Requesting Information About a Location. In *Proceedings of the SIGIR Workshop on Information Retrieval for Question Answering (IR4QA)*, pages 17–22, 2004.

[52] Ralph Grishman and Beth Sundheim. Message understanding conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics*, pages 466–471, Morristown, NJ, USA, 1996. Association for Computational Linguistics.

[53] Sanda Harabagiu and Dan Moldovan. An Intelligent System for Question Answering. In *Proceedings of the 5th International Conference on Intelligent Systems*, pages 71–75, Reno, NV, USA, 1996.

[54] Sanda Harabagiu, Dan Moldovan, Christine Clark, Mitchell Bowden, John Williams, and Jeremy Bensley. Answer Mining by Combining Extraction Techniques with Abductive Reasoning. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of TREC-2003, 12th Text Retrieval Conference*, Gaithersburg, US, 2003. National Institute of Standards and Technology, Gaithersburg, US.

[55] Donna Harman. Overview of TREC-1. In *HLT '93: Proceedings of the workshop on Human Language Technology*, pages 61–65, Morristown, NJ, USA, 1993. Association for Computational Linguistics.

[56] Ulf Hermjakob. Parsing and Question Classification for Question Answering. In *Proceedings of the Workshop on Open-Domain Question Answering at ACL-2001*, Toulouse, France, July 2001.

[57] Andrew Hickl, Kirk Roberts, Bryan Rink, Jeremy Bensley, Tobias Jungen, Ying Shi, and John Williams. Question Answering with LCC's CHAUCER-2 at TREC 2007. In *Proceedings of the TREC 2007*, 2007.

[58] L Hirschman, M Light, E Breck, and J Burger. Deep Read: A reading comprehension system. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 325–332, 1999.

[59] Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, and Chin-Yew Lin. Question Answering in Webclopedia. In Ellen M. Voorhees and Donna K. Harman, editors, *Proceedings of TREC-9, 9th Text Retrieval Conference*, pages 655–654, Gaithersburg, US, 2000. National Institute of Standards and Technology, Gaithersburg, US.

[60] K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, and Y. Wilks. University of sheffield: Description of the LASIE-II system as used for MUC-7. In *Proceedings of the Seventh Message Understanding Conferences (MUC-7)*. Morgan Kaufman, 1998.

[61] Paul Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

[62] Valentin Jijkoun, Maarten de Rijke, and Jori Mur. Information extraction for question answering: improving recall through syntactic patterns. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 1284, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[63] Karen Spärck Jones and Julia R. Galliers. *Evaluating Natural Language Processing Systems: An Analysis and Review*. Springer-Verlag New York, Inc., 1996.

[64] Michael Kaisser. The QuALiM question answering demo: Supplementing answers with paragraphs drawn from Wikipedia. In *Proceedings of*

the ACL-08: HLT Demo Session, pages 32–35, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[65] Michael Kaisser, Marti A. Hearst, and John B. Lowe. Improving Search Results Quality by Customizing Summary Lengths. In *Proceedings of ACL-08: HLT*, pages 701–709, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[66] Michael Kaisser and Bonnie Webber. Question Answering based on Semantic Roles. In *Proceedings of the ACL 2007 Workshop on Deep Linguistic Processing*, pages 41–48, Prague, Czech Republic, June 2007. c2007 Association for Computational Linguistics.

[67] Noriko Kando, Kazuko Kuriyama, Toshihiko Nozue, Koji Eguchi, Hiroyuki Kato, and Soichiro Hidaka. Overview of ir tasks at the first ntcir workshop. In *Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*, pages 11–44, August 30 - September 1 1999.

[68] Boris Katz, Sue Felshin, Gregory Marton, Federico Mora, Yuan K. Shen, Gabriel Zaccak, Ammar Ammar, Eric Eisner, Asli Turgut, and L. Brown Westrick. CSAIL at TREC 2007 Question Answering. In *Proceedings of the TREC 2007*, 2007.

[69] Boris Katz, Jimmy Lin, and Sue Felshin. Gathering Knowledge for a Question Answering System from Heterogeneous Information Sources. In *Proceedings of the ACL 2001 Workshop on Human Language Technology and Knowledge Management*, Toulouse, France, 2001.

[70] Boris Katz, Jimmy Lin, Daniel Loreto, Wesley Hildebrandt, Matthew Bilotti, Sue Felshin, Aaron Fernandes, Gregory Marton, and Federico Mora. Integrating Web-based and Corpus-based Techniques for Question Answering. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of TREC-2003, 12th Text Retrieval Conference*,

Gaithersburg, US, 2003. National Institute of Standards and Technology, Gaithersburg, US.

[71] P. Koomen, V. Punyakanok, D. Roth, and W. Yih. Generalized Inference with Multiple Semantic Role Labeling Systems (Shared Task Paper). In Ido Dagan and Dan Gildea, editors, *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*, pages 181–184, 2005.

[72] Cody Kwok, Oren Etzioni, and Daniel S. Weld. Scaling question answering to the web. *ACM Trans. Inf. Syst.*, 19(3):242–262, 2001.

[73] Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics*, pages 556–562, Taipei, Taiwan, 2002. Association for Computational Linguistics.

[74] Jimmy Lin. The Web as a resource for question answering: Perspectives and challenges. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002)*, 2002.

[75] Jimmy Lin. Evaluation of resources for question answering evaluation. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 392–399, New York, NY, USA, 2005. ACM.

[76] Jimmy Lin. An exploration of the principles underlying redundancy-based factoid question answering. *ACM Trans. Inf. Syst.*, 25(2):6, 2007.

[77] Kenneth C. Litkowski. Question-Answering Using Semantic Relation Triples. In *Proceedings of the 8th Text Retrieval Conference (TREC-8)*, pages 349–356, 1999.

[78] Kenneth C. Litkowski. Syntactic clues and lexical resources in question-answering. In *The Ninth Text Retrieval Conference (TREC-*

9). NIST Special Publication 500-249. Gaithersburg, MD, pages 157–166, 2000.

[79] Kenneth C. Litkowski. CL Research Experiments in TREC-10 Question Answering. In *The Tenth Text REtrieval Conference (TREC 2001). NIST Special Publication 500-250. Gaithersburg, MD*, pages 122–131, 2001.

[80] Kenneth C. Litkowski. Question Answering Using XML-Tagged Documents. In *The Eleventh Text Retrieval Conference (TREC 2002) NIST Special Publication: SP 500-251. Gaithersburg, MD*, pages 156–165, 2002.

[81] Carol Lundquist, David A. Grossman, and Ophir Frieder. Improving relevance feedback in the vector space model. In *CIKM '97: Proceedings of the sixth international conference on Information and knowledge management*, pages 16–23, New York, NY, USA, 1997. ACM.

[82] Craig Macdonald and Iadh Ounis. The TREC Blogs06 Collection : Creating and Analysing a Blog Test Collection. Technical Report TR-2006-224, Department of Computing Science, University of Glasgow, 2006.

[83] Bernardo Magnini, Danilo Giampiccolo, Pamela Forner, Christelle Ayache, Valentin Jijkoun, Petya Osenova, Anselmo Peñas, Paulo Rocha, Bogdan Sacaleanu, and Richard F. E. Sutcliffe. Overview of the CLEF 2006 Multilingual Question Answering Track. In Carol Peters, Paul Clough, Fredric C. Gey, Jussi Karlgren, Bernardo Magnini, Douglas W. Oard, Maarten de Rijke, and Maximilian Stempfhuber, editors, *CLEF*, volume 4730 of *Lecture Notes in Computer Science*, pages 223–256. Springer, 2006.

[84] Bernardo Magnini, Simone Romagnoli, Alessandro Vallin, Jesús Herrera, Anselmo Peñas, Víctor Peinado, Felisa Verdejo, and Maarten de Rijke. The Multiple Language Question Answering Track at CLEF

2003. In Carol Peters, Julio Gonzalo, Martin Braschler, and Michael Kluck, editors, *CLEF*, volume 3237 of *Lecture Notes in Computer Science*, pages 471–486. Springer, 2003.

[85] Gideon S. Mann. Fine-Grained Proper Noun Ontologies for Question Answering. In *Proceedings of the SemaNet'02 : Building and Using Semantic Networks*, Taipei, Taiwan, 2002.

[86] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, July 2008.

[87] Lluis Marquez, Xavier Carreras, Kenneth C. Litkowski, and Suzanne Stevenson. Semantic Role Labeling: An Introduction to the Special Issue. *Computational Linguistics*, 34(2):145–159, 2008.

[88] Yosi Mass, Matan Mandelbrod, Einat Amitay, David Carmel, Yoëlle S. Maarek, and Aya Soffer. JuruXML - an XML Retrieval System at INEX'02. In Fuhr et al. [45], pages 73–80.

[89] Igor Aleksandrovič Melčuk. *Dependency Syntax: Theory and Practice*. SUNY Press, 1988. ISBN 0887064507, 9780887064500.

[90] Donald Metzler, Trevor Strohman, Howard Turtle, and Bruce W. Croft. Indri at TREC 2004: Terabyte Track. In *Text REtrieval Conference (TREC 2004)*, 2005.

[91] D. Moldovan, C. Clark, S. Harabagiu, and D. Hodges. Cogex: A semantically and contextually enriched logic prover for question answering. *Journal of Applied Logic*, 5(1):49–69, 2007.

[92] Dan Moldovan, Mitchell Bowden, and Marta Tatu. A Temporally-Enhanced PowerAnswer in TREC 2006. In *The Fifteenth Text REtrieval Conference (TREC 2006)*, 2006.

[93] Dan Moldovan, Christine Clark, and Mitchell Bowden. Lymba's PowerAnswer 4 in TREC 2007. In *Proceedings of the TREC 2007*, 2007.

[94] Dan Moldovan, Marius Paşca, Sanda Harabagiu, and Mihai Surdeanu. Performance issues and error analysis in an open-domain question answering system. *ACM Trans. Inf. Syst.*, 21(2):133–154, 2003.

[95] Diego Mollá. AnswerFinder in TREC 2003. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of TREC-2003, 12th Text Retrieval Conference*, Gaithersburg, US, 2003. National Institute of Standards and Technology, Gaithersburg, US.

[96] Diego Mollá and Menno van Zaanen. AnswerFinder at TREC 2005. In *The Fourteenth Text REtrieval Conference (TREC 2005)*, Gaithersburg, Maryland, 2006. National Institute of Standards and Technology.

[97] Christof Monz. *From Document Retrieval to Question Answering*. PhD thesis, University of Amsterdam, 2003.

[98] Christof Monz. Minimal Span Weighting Retrieval for Question Answering. In *Proceedings of the SIGIR-2004 Workshop on Information Retrieval For Question Answering (IR4QA)*, Sheffield, UK, July 2004.

[99] Jack Muramatsu and Wanda Pratt. Transparent Queries: investigation users' mental models of search engines. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 217–224, New York, NY, USA, 2001. ACM.

[100] Vanessa Murdock and W. Bruce Croft. Simple Translation Models for Sentence Retrieval in Factoid Question Answering. In *Proceedings of the SIGIR-2004 Workshop on Information Retrieval For Question Answering (IR4QA)*, pages 31–35, Sheffield, UK, July 2004.

[101] Seung-Hoon Na, In-Su Kang, Sang-Yool Lee, and Jong-Hyeok Lee. Using Grammatical Relations, Answer Frequencies and the World Wide Web for TREC Question Answering. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of TREC-2002, 11th Text Retrieval*

Conference, Gaithersburg, US, 2002. National Institute of Standards and Technology, Gaithersburg, US.

[102] Srini Narayanan and Sanda Harabagiu. Question answering based on semantic structures. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 693, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[103] Gunter Neumann and Bogdan Sacaleanu. Experiments on Robust NL Question Interpretation and Multi-layered Document Annotation for a Cross-Language Question/Answering System. In *Multilingual Information Access for Text, Speech and Images: 5th Workshop of the Cross-Language Evaluation Forum, CLEF 2004, Bath, UK, September 15-17, 2004: Revised Selected Papers*. Springer, 2005.

[104] Scott Nowson and Robert Dale. Charting Democracy Across Parsers. In *Proceedings of the Australasian Language Technology Workshop 2007*, pages 75–82, Melbourne, Australia, December 2007.

[105] Paul Ogilvie and Jamie Callan. Parameter Estimation for a Simple Hierarchical Generative Model for XML Retrieval. *Advances in XML Information Retrieval and Evaluation*, pages 211–224, 2006.

[106] Marius Paşca and Sanda Harabagiu. High Performance Question/Answering. In *Proceedings of SIGIR'01*, New Orleans, Louisiana, USA, 2001. ACM.

[107] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106, 2005.

[108] B. Pan, H. Hembrooke, T. Joachims, L. Lorigo, G. Gay, and L. Granka. Google We Trust: Users' Decisions on Rank, Position, and Relevance. *Journal of Computer-Mediated Communication*, 12(3):801–823, 2007.

[109] Luiz Augusto Pizzato. Using a Trie-based Structure for Question Analysis. Presentation at the Australasian Language Technology Workshop 2004, Macquarie University, Sydney, Australia, 8 December 2004.

[110] Luiz Augusto Pizzato. Using a Trie-based Structure for Question Analysis. In Ash Asudeh, Cécile Paris, and Stephen Wan, editors, *Proceedings of the Australasian Language Technology Workshop 2004*, pages 25–31, Macquarie University, Sydney, Australia, December 2004. ASSTA. ISBN: 0 9581946 1 0.

[111] Luiz Augusto Pizzato and Diego Mollá. Extracting Exact Answers using a Meta Question Answering System. In *Proceedings of the Australasian Language Technology Workshop 2005 (ALTA-2005)*, The University of Sydney, Australia, December 2005.

[112] Luiz Augusto Pizzato, Diego Mollá, and Cécile Paris. Pseudo Relevance Feedback Using Named Entities for Question Answering. In *Proceedings of the Australasian Language Technology Workshop 2006*, Sydney, 2006.

[113] Warren J. Plath. Request: A natural language question-answering system. *IBM Journal of Research and Development*, 20(4):326–335, 1976.

[114] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281, New York, NY, USA, 1998. ACM.

[115] Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James Martin, and Dan Jurafsky. Shallow Semantic Parsing Using Support Vector Machines. In *Proceedings of the Human Language Technology Conference/North American chapter of the Association of Computational Linguistics (HLT/NAACL)*, Boston, MA, 2004.

[116] Dragomir R. Radev, Hong Qi, Harris Wu, and Weiguo Fan. Evaluating Web-based Question Answering Systems. In *Demo section, LREC 2002*, Las Palmas, Spain, June 2002.

[117] Dragomir R. Radev, Hong Qi, Zhiping Zheng, Sasha Blair-Goldensohn, Zhu Zhang, Weiguo Fan, and John Prager. Query Modulation for Web-based Question Answering. In Tomek Strzalkowski and Sanda Harabagiu, editors, *Advances in Open-Domain Question Answering*, volume 32 of *Text, Speech and Language Technology*, pages 285–303. Springer Netherlands, October 2003.

[118] Majid Razmara, Andrew Fee, and Leila Kosseim. Concordia University at the TREC 2007 QA track. In *Proceedings of the TREC 2007*, 2007.

[119] Ian Roberts and Robert J. Gaizauskas. Evaluating Passage Retrieval Approaches for Question Answering. In Sharon McDonald and John Tait, editors, *Advances in Information Retrieval, 26th European Conference on IR Research, ECIR 2004, Sunderland, UK, April 5-7, 2004, Proceedings*, volume 2997 of *Lecture Notes in Computer Science*, pages 72–84. Springer, 2004.

[120] S E Robertson, S Walker, S Jones, M M Hancock-beaulieu, and M Gatford. Okapi at TREC-3. In *The Third Text REtrieval Conference (TREC-3)*, pages 109–126, 1995.

[121] Stephen E. Robertson. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation*, 60(5):503–520, 2004.

[122] Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Aarron Gull, and Marianna Lau. Okapi at TREC. In *Text REtrieval Conference*, pages 21–30, 1992.

[123] Josef Ruppenhofer, Michael Ellsworth, Miriam R. L. Petruck, and Christopher R. Johnson. FrameNet II: Extended Theory and Practice. Technical report, ICSI Technical Report, 2005.

[124] Ian Ruthven and Mounia Lalmas. A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review*, 18(2):95–145, 2003.

[125] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

[126] Gerard Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

[127] Gerard Salton and Chris Buckley. Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[128] Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.

[129] Ashish Kumar Saxena, Ganesh Viswanath Sambhu, L. Venkata Subramaniam, and Saroj Kaushik. Iitd-ibmirl system for question answering using pattern matching, semantic type and semantic category recognition. In *Proceedings of the TREC 2007*, 2007.

[130] Nico Schlaefer, Petra Gieselmann, and Guido Sautter. The Ephyra QA system at TREC 2006. In *The Fifteenth Text REtrieval Conference (TREC 2006)*, 2006.

[131] Nico Schlaefer, Jeongwoo Ko, Justin Betteridge, Guido Sautter, Manas Pathak, and Eric Nyberg. Semantic Extensions of the Ephyra QA System for TREC 2007. In *Proceedings of the Sixteenth Text REtrieval Conference (TREC 2007)*, 2007.

[132] Sam Scott and Robert Gaizauskas. University of Sheffield TREC-9 Q & A System. In Ellen M. Voorhees and Donna K. Harman, editors, *Proceedings of TREC-9, 9th Text Retrieval Conference*, pages 635–634,

Gaithersburg, US, 2000. National Institute of Standards and Technology, Gaithersburg, US.

[133] Dan Shen and Dietrich Klakow. Exploring correlation of dependency relation paths for answer extraction. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 889–896, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[134] Dan Shen and Mirella Lapata. Using Semantic Roles to Improve Question Answering. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 12–21, Prague, June 2007, 2007. Association for Computational Linguistics.

[135] Dan Shen, Michael Wiegand, Andreas Merkel, Stefan Kazalski, Sabine Hunsicker, Jochen L. Leidner, and Dietrich Klakow. The Alyssa System at TREC QA 2007: Do We Need Blog06? In *Proceedings of the TREC 2007*, 2007.

[136] Robert F. Simmons. Answering English questions by computer: a survey. *Commun. ACM*, 8(1):53–70, 1965.

[137] Mark D. Smucker, James Allan, and Ben Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 623–632, New York, NY, USA, 2007. ACM.

[138] Trevor Strohman, Donald Metzler, Howard Turtle, and W. Bruce Croft. Indri: A language-model based search engine for complex queries (extended version). IR 407, CIIR Technical Report, University of Massachusetts, 2005.

[139] Tomek Strzalkowski. *Natural Language Information Retrieval*. Kluwer Academic Publishers, Norwell, MA, USA, 1999.

[140] Tomek Strzalkowski, Jose Perez Carballo, Pasi Tapanainen, Timo Jarvinen, Jussi Karlgren, and Anette Hulth. Natural Language Information Retrieval: TREC-7 Report. In *Proceedings of the Text REtrieval Conference*, pages 164–173, 1998.

[141] R. X. Sun, J. J. Jiang, Y. F. Tan, H. Cui, T. S. Chua, and M. Y. Kan. Using syntactic and semantic relation analysis in question answering. In *Proceedings of the TREC*, 2005.

[142] Mihai Surdeanu and Jordi Turmo. Semantic Role Labeling Using Complete Syntactic Analysis. In *Proceedings of CoNLL 2005 Shared Task*, June 2005.

[143] Pasi Tapanainen and Timo Jarvinen. A non-projective dependency parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 64–71, 1997.

[144] Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 41–47, New York, NY, USA, 2003. ACM Press.

[145] Jörg Tiedemann. Improving Passage Retrieval in Question Answering Using NLP. In *Chapter 9 - Text Mining and Applications (TEMA 2005)*, volume 3808/2005 of *Progress in Artificial Intelligence*, pages 634–646. Springer Berlin / Heidelberg, 2005.

[146] Jörg Tiedemann. Integrating linguistic knowledge in passage retrieval for question answering. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural*

Language Processing, pages 939–946, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[147] Jörg Tiedemann. Optimizing information retrieval in question answering using syntactic annotation. In *Proceedings of RANLP 2005*, pages 540–546, 2005.

[148] Jörg Tiedemann. A Comparison of Genetic Algorithms for Optimizing Linguistically Informed IR in Question Answering. In *Proceedings of the AI*IA 2007 Artificial Intelligence and Human-oriented Computing: 10th Congress of the Italian Association for Artificial Intelligence, Rome, Italy, September 10-13, 2007*. Springer, 2007.

[149] David Tomás, José L. Vicedo, Maximiliano Saiz, and Rubén Izquierdo. An XML-Based System for Spanish Question Answering. *Part IV. Multiple Language Question Answering (QA@CLEF). Accessing Multilingual Information Repositories. Lecture Notes in Computer Science*, 4022/2006:347–350, October 2006.

[150] Andrew Trotman. Compressing Inverted Files. *Information Retrieval*, 6(1):5–19, 2003.

[151] Nicolas Usunier, Massih R. Amini, and Patrick Gallinari. Boosting Weak Ranking Functions to Enhance Passage Retrieval For Question Answering. In *IR4QA: Information Retrieval for Question Answering Workshop. SIGIR'04*, 2004.

[152] Lonneke van der Plas and Jörg Tiedemann. Using Lexico-Semantic Information for Query Expansion in Passage Retrieval for Question Answering. In *Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question Answering*, pages 50–57, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

[153] C. J. van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 1979.

[154] Ellen M. Voorhees. The TREC-8 Question Answering Track Report. In *Proceedings of TREC-8*, pages 77–82, 1999.

[155] Ellen M. Voorhees. Overview of the TREC-9 Question Answering Track. In *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*, pages 71–80, 2000.

[156] Ellen M. Voorhees. Overview of the TREC 2001 Question Answering Track. In *Proceedings of The Tenth Text REtrieval Conference (TREC 2001)*, pages 42–51, 2002.

[157] Ellen M. Voorhees. Overview of the TREC 2002 Question Answering Track. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of TREC-2002, 11th Text Retrieval Conference*, Gaithersburg, US, 2002. National Institute of Standards and Technology, Gaithersburg, US.

[158] Ellen M. Voorhees. The philosophy of information retrieval evaluation. In *Proceedings of the The Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems*, pages 355–370. Springer-Verlag, 2002.

[159] Ellen M. Voorhees. Overview of the TREC 2003 Question Answering Track. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of TREC-2003, 12th Text Retrieval Conference*, Gaithersburg, US, 2003. National Institute of Standards and Technology, Gaithersburg, US.

[160] Ellen M. Voorhees. Overview of TREC 2003. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of TREC-2003, 12th Text Retrieval Conference*, Gaithersburg, US, 2003. National Institute of Standards and Technology, Gaithersburg, US.

[161] Ellen M Voorhees. Overview of the TREC 2004 Question Answering Track. In *Text REtrieval Conference*, 2004.

[162] Ellen M. Voorhees and Hoa Trang Dang. Overview of the TREC 2005 Question Answering Track. In *Text REtrieval Conference*, 2005.

[163] Ellen M. Voorhees and Dawn M. Tice. The TREC-8 question answering track evaluation. In *Text Retrieval Conference TREC-8*, pages 83–105, 1999.

[164] Ellen M. Voorhees and Dawn M. Tice. Building a question answering test collection. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207, Athens, Greece, 2000. ACM Press.

[165] Kieran White and Richard F. E. Sutcliffe. Seeking an Upper Bound to Sentence Level Retrieval in Question Answering. In *IR4QA: Information Retrieval for Question Answering Workshop. SIGIR '04*, 2004.

[166] Edward W. D. Whittaker, Matthias H. Heie, Josef R. Novak, and Sadaoki Furui. TREC 2007 Question Answering Experiments at Tokyo Institute of Technology. In *Proceedings of the TREC 2007*, 2007.

[167] Kevin Wilkinson, Craig Sayers, Harumi A. Kuno, and Dave Reynolds. Efficient rdf storage and retrieval in jena2. In Isabel F. Cruz, Vipul Kashyap, Stefan Decker, and Rainer Eckstein, editors, *SWDB*, pages 131–150, 2003.

[168] Hugh E. Williams, Justin Zobel, and Dirk Bahle. Fast phrase querying with combined indexes. *ACM Transactions on Information Systems*, 22(4):573–594, 2004.

[169] Jinxi Xu, Ralph Weischedel, and Chanh Nguyen. Evaluating a probabilistic model for cross-lingual information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 105–110, New York, NY, USA, 2001. ACM.

[170] Menno Van Zaanen, Diego Mollá, and Luiz Augusto Pizzato. AnswerFinder at TREC 2006. In *The Fifteenth Text REtrieval Conference (TREC 2006)*, 2007.

[171] Menno Van Zaanen, Luiz Augusto Pizzato, and Diego Mollá. Classifying Sentences using Induced Structure. In *Proceedings of the Twelfth Symposium on String Processing and Information Retrieval (SPIRE-2005)*, Buenos Aires, Argentina, November 2005.

[172] Menno Van Zaanen, Luiz Augusto Pizzato, and Diego Mollá. Question Classification by Structure Induction. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-2005)*, Edinburgh, Scotland, August 2005.

[173] Dell Zhang and Wee Sun Lee. A Web-based Question Answering System. In *Proceedings of the SMA Annual Symposium 2003*, Singapore, 2003.

[174] Dell Zhang and Wee Sun Lee. Question classification using support vector machines. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (SIGIR-03)*, pages 26–32. ACM Press, 2003.

[175] Zhiping Zheng. AnswerBus Question Answering System. In *Human Language Technology Conference (HLT 2002)*, San Diego, CA, March 24-27 2002.

[176] George Kingley Zipf. *The Psycho-Biology of Language.* MIT Press, Cambridge, Massachusetts, USA, 1935.

[177] Justin Zobel. How reliable are the results of large-scale information retrieval experiments? In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 307–314, New York, NY, USA, 1998. ACM.

[178] Justin Zobel, Alistair Moffat, and Kotagiri Ramamohanarao. Inverted files versus signature files for text indexing. *ACM Trans. Database Syst.*, 23(4):453–490, 1998.