Olga Marina Freitas Craveiro

Segment-Based Temporal Information Retrieval

Tese de doutoramento do Programa de Doutoramento em Ciências e Tecnologias da Informação, orientada pelo Professor Doutor Joaquim Melo Henriques de Macedo e pelo Professor Doutor Henrique Santos do Carmo Madeira, e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Setembro de 2015

· U C ·

Universidade de Coimbra

Olga Marina Freitas Craveiro

Segment-Based Temporal Information Retrieval

Thesis submitted to the University of Coimbra for the degree of Philosophiae Doctor (PhD)
in Information Science and Technology

University of Coimbra
Faculty of Sciences and Technology
Department of Informatics Engineering
September, 2015

Dissertation conducted under the supervision of

Professor Joaquim Melo Henriques de Macedo

Assistant Professor of
Department of Informatics, University of Minho

Professor Henrique Santos do Carmo Madeira

Full Professor of
Department of Informatics Engineering
Faculty of Sciences and Technology, University of Coimbra

This work is partially sponsored by Fundação para a Ciência e a Tecnologia (FCT), under the grant SFRH/BD/48583/2008.

POPH QUALIFICAR É CRESCER.

QREN QUADRO DE REFERÊNCIA ESTRATÉGICO NACIONAL PORTUGAL 2007.2013

GOVERNO DA REPÚBLICA PORTUGUESA

UNIÃO EUROPEIA
Fundo Social Europeu

“How can he recognize that his work is finished? That's a decision he has to take… In fact, the completion of a work is only ever an abandonment, a halt that can always be regarded as fortuitous in an evolution that might have been continued.”

Paul Valéry (1871 – 1945)

To my beloved parents

Alfredo José (in memorium)

Marília Manuela

Abstract

The Web is actually the key information source for our daily lives. Search engines are essential to use efficiently the information available at the Web. Therefore, there is an intensive academic and industrial research effort to improve the efficiency and effectiveness of underlying Web information retrieval models.

Temporal information plays an important role for text understanding, allowing the identification of relations between entities, facts or events described by documents. Besides that, the time dimension is also an important element in the context of the user's information need, and if used carefully it can improve the effectiveness of search applications.

Indeed, temporal information can be a key piece on most information system applications and, consequently, in Web based applications, since temporal information can be found in every document, either with the metadata, such as the creation or publication date, or in the document content in the form of temporal references, such as dates and time.

Recognizing temporal information and putting such information in a machine-readable format is the starting point for these systems to take advantage of it, improving their functionalities and adding new features. So, the extraction of temporal information from text documents is becoming increasingly important in many applications, such as natural language processing, information retrieval, question answering, etc.

Initially, this research was concerned with improving the quality of the results, incorporating temporal information in information retrieval systems using Portuguese texts; such information is not reduced to document timestamps, including also the time extracted from the content itself. However, working with the Portuguese language was one of the greatest challenges faced due to the lack of resources, namely corpora and software, which led us to create the instruments needed for the research throughout the course of the work. For this reason, the research was not only focused on document retrieval, but also covers the development of tools to process Portuguese texts.

In this thesis, we propose an original method for easy recognition of temporal expressions in Portuguese texts. The method creates semantically classified temporal patterns, based on regular expressions, by using word co-occurrences obtained from corpora and a pre-defined seed keywords set, which were derived from the temporal references of the used language.

In order to have a temporal machine-readable representation of documents, after the recognition of temporal expressions it is required to capture the normalized time values, when possible. We propose an approach for the resolution of temporal expressions, achieving promising results in a Portuguese collection.

Our proposal for the time-aware model takes advantage of temporal discontinuities in text to establish a relationship between time and document terms. Since words often describe facts and events, this relationship allows a better understanding of the texts and provides a more effective extraction of implicit or explicit temporal information. By using the segmentation of texts based on temporal discontinuities, the indexes can be enriched with temporal information, improving the effectiveness of information retrieval systems, for example.

This work represents a step forward for Portuguese language processing, with a notorious lack of tools. Even with target application in time-aware information retrieval, the proposed tools for the processing of the Portuguese language can be used in other application scenarios.

Keywords: Temporal Information Retrieval, Temporal Information Extraction, Temporal Text Segmentation

Resumo

A Web é, na verdade, uma fonte de informação fundamental utilizada no nosso dia-a-dia. Os motores de busca são essenciais para aceder de forma eficiente à informação disponível na Web. Assim, nos últimos anos tem sido realizada muita investigação, quer no meio académico quer no meio empresarial, para o melhoramento da eficiência e da eficácia dos modelos de recuperação de informação na Web.

A informação temporal tem um papel importante na compreensão de textos, permitindo a identificação de relações entre entidades, factos ou eventos descritos pelos documentos. Para além disso, a dimensão temporal é também um elemento importante no contexto das necessidades de informação dos utilizadores e, se for utilizada de forma eficaz, pode melhorar o desempenho dos sistemas de recuperação de informação.

Na verdade, a informação temporal pode ser uma peça chave na maioria das aplicações de sistemas de informação e, consequentemente nas aplicações Web, uma vez que a informação temporal pode ser encontrada em todos os documentos, quer nos metadados, tais como as datas de criação, atualização ou publicação dos documentos, quer sob a forma de referências temporais existentes no conteúdo dos documentos, de forma explícita ou implícita.

O reconhecimento de tal informação e a sua colocação num formato reconhecido pelos sistemas é o ponto de partida para que estes sistemas possam utilizá-la no melhoramento de funcionalidades já existentes, ou até mesmo disponibilizando outras funcionalidades. Por isso, a extração de informação temporal dos documentos de texto tem-se tornado cada vez mais importante em muitas aplicações, como por exemplo, processamento de linguagem natural, recuperação de informação, sistemas de pergunta-resposta, etc.

Este trabalho de investigação começou por centrar-se na melhoria da qualidade dos resultados de sistemas de recuperação de informação que processam texto em língua Portuguesa, através da incorporação de informação temporal, considerando não só a marca temporal dos documentos, como também as referências temporais extraídas do conteúdo dos documentos. No entanto, o trabalho com a língua Portuguesa foi um dos maiores desafios encontrados devido, principalmente, à falta de recursos, nomeadamente *corpora* para a realização de testes experimentais e software para o seu processamento. Assim, fomos obrigados a criar os recursos (ferramentas e *corpora*) que foram sendo necessários ao longo do trabalho. Por este motivo, a investigação não ficou somente focada na recuperação de informação, estendendo-se também o desenvolvimento das ferramentas necessárias ao processamento de textos em língua Portuguesa.

Nesta tese apresentamos um método original que permite o reconhecimento de expressões temporais em textos escritos em Português, recorrendo a um algoritmo simples e de fácil processamento. O método cria padrões temporais classificados semanticamente utilizando expressões regulares. A sua criação é feita recorrendo à co-ocorrências de palavras obtidas a partir de vários corpora de treino e de um conjunto predefinido de palavras-chave. Palavras essas que são extraídas das referências temporais existentes na língua utilizada, que neste caso é o Português.

Por forma a chegarmos a uma representação temporal dos documentos que tem de ser compreendida pelos sistemas, depois do reconhecimento das expressões temporais é necessário realizar a normalização dos valores temporais, sempre que isso seja possível. É proposta uma abordagem para a resolução das expressões temporais que nos testes experimentais realizados numa coleção com documentos em Português atingiu resultados muito promissores.

A nossa proposta para o modelo de recuperação de informação com dimensão temporal aproveita as descontinuidades temporais do texto para estabelecer uma relação entre as referências temporais, no formato de datas devidamente normalizadas, e os termos do documento. Visto que as palavras descrevem frequentemente factos e eventos, esta relação permite obter um maior conhecimento dos textos e também uma extração mais eficiente da informação temporal implícita ou explícita. Os índices podem ser enriquecidos com a informação temporal através da segmentação de texto baseada nas descontinuidades temporais, e assim, melhorar a eficácia dos sistemas de recuperação de informação.

Este trabalho representa um progresso no processamento da língua Portuguesa onde a falta de recursos é notória. As ferramentas de processamento da língua Portuguesa apresentadas foram construídas com o objetivo de serem usadas pelos sistemas de recuperação de informação com dimensão temporal, embora possam ser aplicadas em outros cenários.

Palavras-Chave: Recuperação de Informação Temporal, Extração de Informação Temporal, Segmentação Temporal de Textos

Acknowledgments

First and foremost, I would like to thank my two supervisors Professor Joaquim Macedo and Professor Henrique Madeira for having enough faith in my skills to give me this opportunity, their support, feedback, and encouragement during this thesis. I am also grateful to Joaquim Macedo for the so useful and important discussions and by much of his spending time in our long meetings on my journeys at the University of Minho.

I want to thank Beatriz Piedade, Cristina Chuva, Filipe Neves, Miguel Frade and Patrício Domingues for encouragement and useful hints; Rosa Matias and Rui Oliveira for making easy the hard task of managing the course of Databases. Special thanks to Catarina Reis for the many and great contributions given during the Java programming tasks.

I would also like to thank all my friends for all relaxing and amusing moments, which were a whiff of fresh air, so important, to clean the mind and encourage me to keep working.

In addition, I am greatly indebted for friendship and the great hospitality during all my journeys in Braga given by my cousins who live there, António, Ermelinda, Ondina and her family.

A special thank to my special friends *Nheu Quaresma* family for friendship, inspiration and always being there when I needed.

Last, but not least, I want to thank my family, specially my mother, for their love, moral support and endless patience during this long journey. This one is for you too.

Table of Contents

List of Figures

List of Tables

Abbreviations and Acronyms

Bo1	Bose-Einstein 1
Bo2	Bose-Einstein 2
COP	Co-Occurrence Processor
CSS	Context-Scanning Strategy
DFR	Divergence From Randomness models
DTD	Document Type Definition
HC	Second HAREM Collection
IMP	Improvement metric
IR	Information Retrieval
MAP	Mean Average Precision
NER	Named Entity Recognition
NLP	Natural Language Processing
PorTexTO	PORtuguese Temporal EXpressions TOol
QA	Question Answering
RI	Robustness Index
SGML	Standard Generalized Markup Language
tmpQE	Temporal Query Expansion
Terrier	TERabyte RetrIEveR
TimeML	Time Markup Language
TmpF	Temporal Filtering
TmpR	Temporal aware Query Reweighting
TmpW	Temporal Weighting
URL	Uniform Resource Locator
UTF	Unicode Transformation Format

XML	EXtensible Markup Language

Chapter 1

Introduction

This thesis addresses the challenge of improving the search results of information retrieval systems taking into account the temporal information found in documents. This chapter explains the motivation of our work and details the definition of the problem we set out to solve. The main contributions of this work are some novel approaches that cross the areas of natural language processing and information retrieval. The organization of the thesis is outlined at the end of this chapter.

1.1 Introduction

Nowadays, searching is the most important activity on the Web, which is used by people as the first step to look for information. Everyday, thousand million queries are executed in Web search engines, making this type of information retrieval systems the fundamental tool used to find information mainly on the Web, but also in other smaller networks. The improvement of the efficiency and effectiveness of information retrieval models is the focus of intensive research.

The motivation of this work is the use of temporal information to improve the search results of information retrieval systems. Time is an important dimension for understanding the text information. However, there is still much to do to achieve its full integration in the most popular retrieval models [Alonso et al., 2011], since the majority of current proposed models only use the creation, modification or publication date of documents and do not take advantage of the richer information in the temporal references available in the content of the documents.

Thus, this work is focused on the design, implementation and evaluation of a time-aware information retrieval model, which is based on the extraction of temporal expressions from the content of document; obviously, we also consider the time metadata of documents.

Unlike other time-aware information retrieval models reported in literature, our approach is based on the temporal segmentation of the text that partitions the text, taking into account the temporal discontinuities of the documents. In this way, the temporal segmentation provides the establishment of a relationship between words and time. So, the words can be temporally related, making our interpretation of texts richer. As the words describe events, facts and named entities, the semantics of the texts can be extracted more naturally and more effectively.

Furthermore, the relationship obtained from temporal segmentation of text is used to enrich the temporal indexes. Although time is often vague or incomplete, a significant part of words and expressions can be placed into a timeline. Then, the information retrieval system can return more effective results to temporal explicit or implicit queries, by using the temporal indexes.

As we intend to work with collections of Portuguese documents, our first concern was to find an available set of tools to process temporal expressions that allow us to reach the focused subject related tasks as soon as possible. However, we found almost no tools for extracting temporal information from texts in Portuguese. The few tools we found were either not available or not appropriate for the purposes. Given this scenario, we developed a toolset from scratch.

When we take the challenge of working with the Portuguese language, we knew it would be a hard work, and this really was the biggest challenge we faced, due to lack of resources. So, the development of the tools to process Portuguese texts became the second focus of this work.

1.2 Problem Definition

The objective of this work is to increase the quality of the results of information retrieval systems by incorporating temporal information in the retrieval models, using temporal attributes in a segment-based document approach. Note that these systems must process Portuguese texts.

The most important research work already carried out so far is based on models for which the words and time are considered independent. This means that although the words and temporal expressions (e.g. extracted dates) occur in the same text, it is not established direct relationships between them. The advantage of such approaches is the simplicity and the ready use of the existing models of information retrieval. The segment-based document approach proposed is specifically intended to explore these relationships using temporal attributes.

Most documents report events and related facts which occur during given periods of time. These periods are established by temporal references in the content of documents. In Web documents, additional temporal references can derive from publication dates of linked documents. So, by obtaining all dates from a document, we can associate each date with a set of neighboring words occurring in the document. From this perspective, the document can be segmented into temporal slices. To achieve the goal of this research, we have faced several challenges, some of them related to natural language processing and others to information retrieval.

The first challenge is the extraction of temporal information from the content of documents and makes it explicit for further processing. In general, collections have a lot of document contents with a large variety of temporal information, besides the time related metadata of documents, also known as document timestamps. In some of these documents, there are temporal expressions with explicit dates and other expressions from which more dates can derived. It is not an easy task to obtain these dates, because some of them are resolved using a date previously referred somewhere in the text and others by only using the document timestamp.

Unlike the English language, the topic of temporal information extraction in Portuguese texts has not been much explored. So, there is a lack of resources, both software and corpora; indeed, there are no publicly available tools to make the extraction of temporal information, as well as to evaluate the tools developed for such purpose.

Since the dates resolved are references to events or actions described in the documents, the sequence of words in the document can be broken in subsequences anchored by the most relevant dates included in the document; therefore, the document can be temporally segmented. A temporal segment can be defined as a fragment of text that does not exhibit abrupt changes in temporal focus [Bramsen et al., 2006]. In addition, the temporal segments in our approach also contain a segment timestamp. Figure 1.1 displays an example of a text to be segmented. The temporal segments of this text are represented in Figure 1.2. The thick rectangle shows the document timestamp.

For information retrieval systems we can consider that there are two types of queries: time-sensitive and time-insensitive. Time-sensitive queries are queries that exhibit, in an explicit or implicit way, a temporal sensitivity and from some of them, we can even extract dates. For example, the queries *Lisboa terramoto*[1] and *Lisboa 1755* are time-sensitive, with implicit and explicit temporal information, respectively. These two queries are focused in the same time period, since there is only one big earthquake occurred in Lisbon on *1st November, 1755*. Indeed, there were other earthquakes, but they are not very important. This particular earthquake is the most famous, because it was the most terrible and disastrous that have occurred in Lisbon. So, if the query *Lisboa terramoto* is submitted in a search engine, the results are documents that talk about the great Lisbon earthquake of 1755. In time-insensitive queries, the user's need does not have obvious time-specific information and it was not found any temporal attributes in the result set. An example is the query *Portugal capital*, where there is not any time reference associated.

In particular, for time-sensitive queries, the temporal segmentation of the document may be used to improve retrieval results, based on ranking formulas considering the time dimension of queries. In the approach proposed, the set of documents retrieved are re-ranked by introducing a temporal factor which fits the relevance score of results to the expected reference of time.

Although, there is work reported in literature about the usage of temporal information for time-sensitive queries [Diaz and Jones, 2004, Dakka et al., 2008, Kanhabua and Nørvåg, 2012, Campos et al., 2014b, Lin et al., 2014, Gupta and Berberich, 2014, Gupta and Berberich, 2015], nevertheless, none of them explores the relationship between words and time given by the temporal segmentation of documents, to the best of our knowledge.

For a temporal rich document along with word position in the text, we can also define a temporal location for words in the text. Therefore, there is a temporal and a position distance between words in the text and these two types of distances are combined.

[1] English version: *Lisbon earthquake*

Trishna e Krishna nasceram no Bangladesh, **há três anos**, unidas pela cabeça. Foram operadas **há um mês** na Austrália e tiveram uma recuperação que surpreendeu os médicos.

As duas meninas do Bangladesh, que **há cinco semanas** ainda estavam unidas pela cabeça, saíram **hoje**, **segunda-feira**, do hospital pediátrico em Melbourne onde foram separadas.

As gémeas registaram uma recuperação que surpreendeu a equipa médica que as operou durante 32 horas e que tem acompanhado a sua situação desde que chegaram à Austrália, **há dois anos**.

Trishna e Krishna tinham apenas 25 por cento de hipóteses de sobreviverem à cirurgia e aparentemente não foram detectados quaisquer danos cerebrais.

Agora, as meninas preparam-se para festejar o seu terceiro aniversário, **amanhã**, **terça-feira**, com uma esperança de vida renovada.

English Version

Trishna and Krishna were born joined at the head in Bangladesh **three years ago**. They were operated in Australia **one month ago** and had a recovery that surprised doctors.

The two girls from Bangladesh, who **five weeks ago** were still joined at the head, came out **today**, **Monday**, the pediatric hospital in Melbourne where they were separated.

The twins showed a recovery that surprised the medical team that operated for 32 hours and that has accompanied their situation since they arrived in Australia **two years ago**.

Trishna and Krishna had only 25 percent chance to survive the surgery and apparently no detected brain damage.

Now, the girls are preparing to celebrate its third anniversary **tomorrow**, **Tuesday**, with a hope of renewed life.

Figure 1.1: News published in the Portuguese newspaper *Jornal de Notícias* on 2009-12-21.

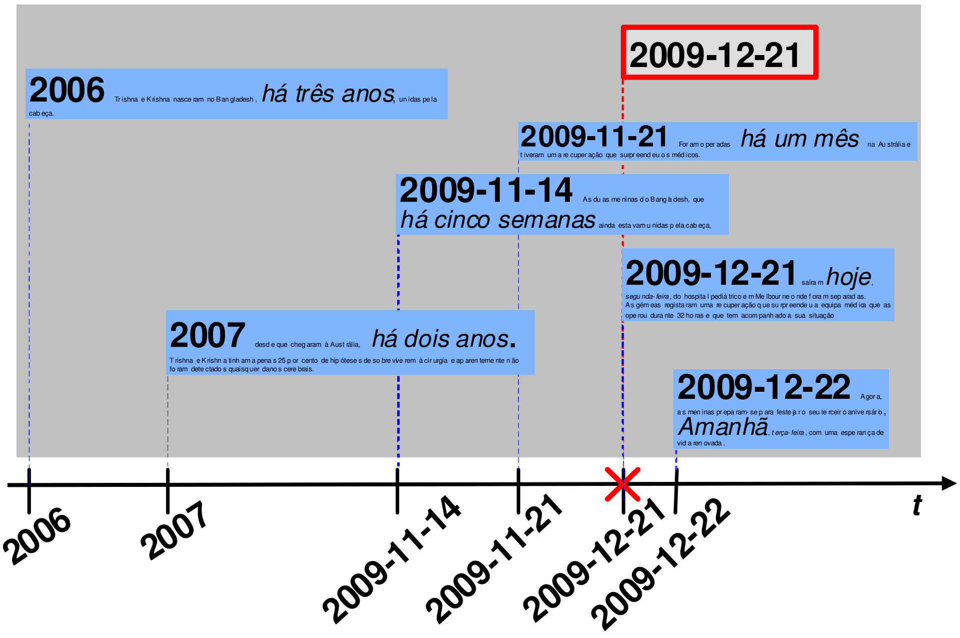

2009-12-21
2006 Trishna e Krishna nasceram no Bangladesh, há três anos, unidas pela cabeça.
2009-11-21 Foram operadas há um mês na Austrália e tiveram uma recuperação que surpreendeu os médicos.
2009-11-14 As duas meninas do Bangladesh, que há cinco semanas ainda estavam unidas pela cabeça,
2009-12-21 saíram hoje, segunda-feira, do hospital pediátrico em Melbourne onde foram separadas. As gémeas registaram uma recuperação que surpreendeu a equipa médica que as operou durante 32 horas e que tem acompanhado a sua situação
2007 desde que chegaram à Austrália, há dois anos. Trishna e Krishna tinham apenas 25 por cento de hipóteses de sobreviverem à cirurgia e aparentemente não foram detectados quaisquer danos cerebrais.
2009-12-22 Agora, as meninas preparam-se para festejar o seu terceiro aniversário, Amanhã, terça-feira, com uma esperança de vida renovada.
2006
2007
2009-11-14
2009-11-21
2009-12-21
2009-12-22
t

Figure 1.2: Temporal segmentation of a document.

When the temporal location of words is used to establish a distance between them, temporal information can be used, even for time-insensitive queries. So, the ranking formulas based on word proximity incorporate time information or instead only use temporal distances. For instance, an example of such time-aware approach is the penalization of inter-segment occurrence for a given set of query terms. This means that inter-segment query terms describe events or facts occurring at different dates.

1.3 Contributions

The objective of this work is to integrate the time dimension in search engines by taking advantage of the temporal characteristics underlying the texts to improve the retrieved documents set. To achieve this goal, we proposed some novel approaches that cross two important areas of research: Natural Language Processing (NLP) and Information Retrieval (IR), namely Temporal Information Retrieval, a relatively new topic of research. To the best of our knowledge, there are no works on time-aware approaches of information retrieval focused on the Portuguese language or works taking into account a temporal segmentation of documents.

Therefore, the main contributions of this work are:

1. A novel approach for time-aware information retrieval based on the assumption that words with the same temporality make the correspondent documents more relevant to the query. Such documents must have a higher rank in the retrieved result set. In addition, we propose five methods of query expansion to improve effectiveness retrieval, which are based on this approach, and evaluate three of them (see Chapters 5 and 6).

2. An approach to extract temporal information from Portuguese texts using semantically classified temporal patterns. Starting from a text document, the result is a set of dates and times normalized in a predefined format. Based on this approach, we developed the *Extraction* tool. Each module of this tool was carried out separately from the others. Chapter 3 describes the proposed approach and each module of the tool. The results obtained of the evaluation are also presented in this chapter.

3. An approach to segment the text into temporally coherent segments, providing the relationship between words and time found in each document. This relationship is crucial to our time-aware information retrieval approach. We also present the *Segmentation* tool that uses the result set obtained by the extraction tool to temporally segment Portuguese

texts. This tool was also evaluated separately from the extraction tool. Chapter 3 explains the algorithm and the evaluation carried out.

4. A set of collections was created to allow the evaluation of the proposed approaches, since there are no test collections for some of this system purposes. The lack of resources to evaluate this kind of systems, when working with the Portuguese language, introduced an increased complexity in the system evaluation task.
5. A temporal characterization of the two collections, Second HAREM and CHAVE, with documents in Portuguese. It gave important findings and increased our sensibility for research in temporal language processing of Portuguese texts (see Chapter 4).

A version of the tools and the collections created during this work are available at https://sites.google.com/site/olgacraveiro/home.

1.4 Structure of the thesis

The thesis is composed of seven chapters.

This chapter introduces the motivation and the definition of the research problem addressed in the thesis. The main contributions of this work are also presented in the present chapter.

Chapter 2 introduces the main aspects of our work and provides an overview of the issues of dealing with time in information retrieval, giving the necessary contextualization of temporal information retrieval for a better understanding of our contributions.

Chapter 3 presents the proposed two approaches in order to obtain the temporal relationships between words found in documents, which are crucial required to the proposal of time-aware temporal retrieval. One approach is to extract relevant temporal information from Portuguese documents. The other approach is to temporally segment texts, considering the time discontinuities found in the content of documents. Based on these approaches, we developed a toolset. The evaluation sessions carried out of each tool and the results obtained are also presented and discussed in this chapter.

The two text collections used in this work, Second HAREM and CHAVE, and their temporal characterization are covered in Chapter 4.

Chapter 5 explains the proposed approach for time-aware information retrieval based on temporal text segmentation. This proposal was applied in two different contexts: Web crawling and retrieval systems when using query expansion technique.

Chapter 6 presents the results obtained in the experiences carried out to validate the temporal query expansions methods.

Finally, Chapter 7 presents the conclusions and future research trends.

Chapter 2

Time in Information Retrieval: An overview

This chapter presents the necessary contextualization of the information retrieval field, for better understanding of our contributions. Some approaches in which time is exploited to benefit different information retrieval tasks are also presented. We also address temporal information extraction and text segmentation, due to their importance for the required temporal retrieval preprocessing in order to establish temporal relationships between words.

2.1 Introduction

Time is naturally associated to documents, either with the metadata, such as the creation or publication date, or in the document content in the form of temporal references. Indeed, any expression that refers to any span of time can be considered a temporal expression. It can be expressed in different ways and represents different time units, such as *"Christmas 2014"*, *"the end of the day"*, *"today"*, *"2014-12-25"*, *"21:00"*, *"from April to May"*, *"during two days"*, *"daily"*, etc.

Time dimension can be a key piece for text understanding, allowing the identification of relations between entities, facts or events described by documents. Besides that, it can also be important in search applications, since time is also present in the context of the user's information need. In fact, temporal information can be incorporated in various contexts as we show in Section 2.3.

Temporal information retrieval models explore time of different ways. Some models only take into consideration the document timestamp and other models also use temporal references found in the content of queries and documents. For these latter models, a temporal retrieval preprocessing must be carried out in both documents and queries, in order to obtain time in a normalized and machine-readable format. Temporal references that cannot be normalized are not incorporated in the retrieval model.

In information retrieval, a text processing is also carried to identify the meaningful words of documents. Thus, documents in temporal retrieval models are modeled as a *bag of normalized temporal expressions* and a *bag of words*. As these models do not establish any relationship between words and time, there is not any dependence between these two sets, contrary to our work. Note that as our proposed approach takes advantage of the temporal relationships between words to improve the effectiveness of information retrieval systems, the temporal retrieval preprocessing also includes text segmentation based on time discontinuities found in the text, to obtain these relationships.

Next subsections describe the fundamental concepts and techniques of information retrieval to provide an overview of relevant approaches found in the literature and help understanding our contributions presented in subsequent chapters.

2.2 Information Retrieval

Information retrieval is the process of discovering documents whose content has relevant information based on the information needs defined by the user, searching from a collection of documents. The information needs is formulated using one or more keywords, named as *query*.

Several models have been proposed by the scientific community over the years, following different approaches, but all of them are based on the three classic model types: Boolean, vector and probabilistic. A description of the three models is presented in Section 2.2.2.

Since the documents collection can be very large, scanning the text sequentially in searching for a query is a time-consuming task. In order to speed up this task, the information retrieval systems build data structures, called *indexes*. The next section describes how the original texts are transformed into index terms.

2.2.1 Document Indexing

Document indexing is the process of converting the original texts of the document collection into indexes. In this way, documents are represented by a set of index terms or keywords that are extracted from the text of the documents. The keywords give the logical view of the document.

Indexes are data structures built-up in order to speed up the query processing. Before a document can be indexed, a text preprocessing is carried out to reduce the set of representative terms. This task can include removal of punctuation marks and stopwords, stemming and lemmatization. In structured documents, the recognition of the internal structure of a document (title, chapters, sections, etc.) is also performed, since this information can be very useful for text understanding [Manning et al., 2008].

The result of the text preprocessing task is a set of terms, named as *index terms*. They are the terms that will be stored in indexes. Besides the terms, inverted index stores their positions in the documents. Additionally, this index can also contain the term frequency in the documents. Information about documents, such as the document length and identifier, are also stored in indexes [Manning et al., 2008].

Figure 2.1 shows an inverted index example. The terms in index are the meaningful words found in the documents. For each term is associated the documents where it occurs and its positions in the documents. In the example, the term *animal* occurs twice in document 1, in positions 3 and 22 of document 1. This term also occurs in document 3, in positions 5, 15 and 23. The term *zoo* occurs in position 15 of the document 2, and in the positions 2 and 14 of the document 3.

ID	Term	DocID: positionList
1	animal	Doc1: (3, 22); Doc3: (5,15,23)
		...
50	zoo	Doc2: (15); Doc3: (2, 14)

Figure 2.1: Inverted indexes: example.

2.2.2 Document Retrieval

A central problem of information retrieval systems is to identify which documents of a collection are relevant for a given user's request and which ones are not. Typically, the decision is given by a ranking algorithm that assigns a score to each document based on the document relevance. The list of the documents retrieved is ordered and the documents at top are considered the relevant documents for the request entered by the user.

Thus, the main objective of a document retrieval model is to define a scenario that carries out the task of finding the relevant documents from a collection, for an information need defined by the user. Although there are several retrieval models, all of them are based on the same principle – matching between the query submitted by the user in the system and the documents of the indexed collection. Indeed, the logical view defined for the user's request and the documents is used for the relevance matching.

According to Baeza-Yates and Ribeiro-Neto (1999), the retrieval models are characterized by the following quadruple *[D, Q, F, R(qi, dj)]*, where:

- *D* is the set of logical view for the documents of the collection;
- *Q* is the logical view for the user's request, named query;
- *F* is the framework where the representation of the documents, queries and their relationships are;
- *R(qi, dj)* is the ranking function which associates a score with one query ($q_i \in Q$) and a document representation ($d_j \in D$). This score, a real number, is calculated based on the relevance of the document d_j for this query.

Each retrieval model can have different specifications for the document representation, query interpretation, framework, and ranking algorithms employed. The following section presents the boolean, vector and probabilistic models. These models became the reference models in information retrieval, as all models that have been proposed are based on one of them.

In addition to the three basic models, we also discuss Divergence From Randomness (DFR) models based on probabilistic model. Although our work does not apply directly these models proposed by Amati and Van Rijsbergen (2002), they are the basis of the Bose-Einstein 1 query expansion model, which is used in the experiments performed to evaluate our temporal methods. This term-weighting model provides a parameter-free approach, being an alternative to the formula proposed by Rocchio (1971) (see Formula 5.3). Query expansion is dully explained in Section 5.3.

Boolean Model

The Boolean model is considered the simplest information retrieval model [Baeza-Yates and Ribeiro-Neto, 1999]. This model is based on set theory and Boolean algebra, which provides an inherent simplicity in the framework. A query is a logic expression where the terms are joined by the Boolean operators AND, OR and NOT. A document is represented as *bag of words*, which means an unordered list of terms. A binary weighting is used to define for each term (*t*), its presence (*1*) or absence (*0*) in each document (*d*), *i.e.*, $w(t,d) \in \{0,1\}$. Table 2.1 shows an example of a term-document matrix, considering a collection with three documents d_1, d_2 and d_3.

Terms	**Documents**		
	d_1	d_2	d_3
t_A	1	0	1
t_B	1	1	1
t_C	0	1	1

Table 2.1. Term-document matrix: an example.

In the Boolean model, the list of relevant documents for a query is obtained by an exact matching between index terms and query terms without ordering the documents. In other words, the model follows a binary classification of the documents. Documents are relevant (1) or non-relevant (0) for the query, without the calculation of any degree of relevance. In this manner, all the relevant documents are qualified with the same importance for the query, but often in reality, this is not true. Thus, the similarity of a document *d* to the query *q* is also a binary function, *i.e.*, $sim(d,q) \in \{0,1\}$.

Considering the documents $d_1 = \{t_A, t_B\}$, $d_2 = \{t_B, t_C\}$ and $d_3 = \{t_A, t_B, t_C\}$ as presented in the term-document matrix of Table 2.1, given a Boolean query $q = t_A$ AND t_B NOT t_C, the result set is $r = \{d_1\}$. The result set is composed of all documents with the terms t_A, t_B, but without the term t_C.

Despite the simplicity of the Boolean model, the query formulation process is quite difficult, since the logic expressions uses a very precise semantic. The AND operator can be very restrictive because the result list can be a very small set of documents. On the other hand, the OR operator can retrieve a large number of documents which could be a problem for the user to find documents satisfying the information needs. Thus, the effectiveness became very dependent on the user's skills.

Vector Space Model

The vector space model [Salton et al., 1975], also known as vector model, recognizing the limitation of binary weights, defined a framework that allows partial matching by using non-binary weights of the terms in documents and in queries. These term weights assume not only the presence or absence of the term, but also their importance in documents, which means that they are used to compute the degree of relevance of the collection documents for a given query. This degree of relevance is used to rank the retrieved documents in a descending order, allowing the partial matching between each document and the query.

The underlying idea of the vector model is that index terms are represented as coordinates in a multidimensional vector space [Sparck Jones and Willett, 1997], which means that the number of dimensions in vector space is defined by the number of distinct index terms of the collection. So, the query and each document of the collection are represented as vectors. The value of each vector coordinate denotes the weight of the term represented by this coordinate, assuming that terms are linearly independent.

Traditionally, the *tf-idf* formula is used to compute the weight of a term, but there are many other formulae, such as several variations of this formula proposed by Salton and Buckley (1988). The formula *tf-idf* assigns a high weight to a term when the term occurs several times in the document, but rarely in the whole collection. The weight of a term *t* in a document *d* can be computed by using *tf-idf*. It is given by the following formula:

$$\text{tf-idf}(t,d) = tf(t,d) \times idf(t) \tag{2.1}$$

In Formula 2.1, *tf* is the frequency of the term *t* in the document *d* and *idf* is the inverse document frequency of a term *t*. The *tf* gives the importance of a term in a document. So, terms that give a good representation of the document topics must have high values of this weight. In general, *tf* can simply be computed as the number of occurrences of the term *t* in a document *d*, *freq(t,d)*, but there are many variations. For example, the *tf-idf* (Formula 2.1) can use the normalized term frequency (*tfn)* which can be computed as:

$$tfn(t,d) = \frac{freq(t,d)}{\max_j freq(j,d)} \tag{2.2}$$

In Formula 2.2, the maximum is computed over all distinct terms (*j*) of the document *d*.

The *idf* of a term *t* measures its importance in the whole document collection, assigning less importance to the terms with high document frequency (*df*). There are some terms that occur too often in the collection, and for that reason, they become less important for relevance determination. For instance, in a collection composed of articles about education, it is highly likely that the term *education* occurs in almost every document.

Let *N* be the total number of documents in the collection and *df* the document frequency of a term *t* defined as the number of documents in the collection that contain the term. Then, *idf* for a term *t* is given by the following formula:

$$idf(t) = \log \frac{N}{df_t} \tag{2.3}$$

The query processing is based on the comparison of the query vector, composed of terms defined in the query and their weights, with each document vector. The correlation between these vectors gives the degree of relevance between the documents of the collection and the user query. The degree of relevance can be computed using the cosine similarity formula, where the value varies from 0 to +1. Therefore, the smaller the angle between the vectors, the shorter the distance between them. If the angle is approximately 0 degrees, the importance given to the terms of the document and the query is approximately the same. So, the similarity obtains the maximum value (+1).

Figure 2.2 shows a graphical representation of the vector model. Let q be the query, d_1, d_2 and d_3 the documents of the collection, t_A, t_B and t_C the terms of the collection, considering d_1= {t_A, t_B}, d_2 = {t_B, t_C } and d_3 = {t_A, t_B, t_C}. Thus, the terms t_A, and t_B are in the content of document d_1, in the document d_2, there are the terms t_B and t_C , and t_A, t_B and t_C are terms of the document d_3.

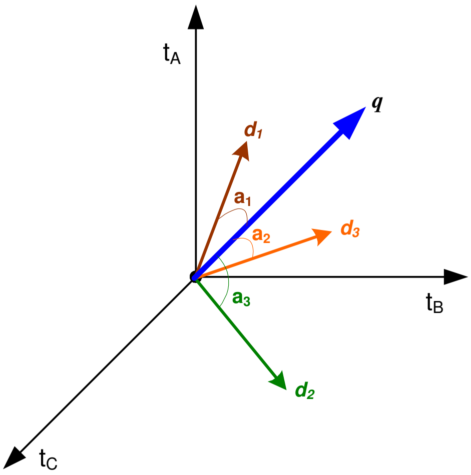


Figure 2.2: Graphical representation of vector model.

All the vectors, documents $(\vec{d}_1, \vec{d}_2, \vec{d}_3)$ and query $\vec{q}$, are represented in a tridimensional space because the collection just have three terms t_A, t_B and t_C . Thus, the cosine of the angle a_1 between the two vectors $\vec{q}$ and $\vec{d}_1$ gives the similarity between the query and the document d_1. The cosine of the angles a_2 and a_3 gives the same information for the documents d_2 and d_3, respectively.

Let $\vec{d}$ be the vector that represents the document and $\vec{q}$ the query vector. The similarity of the two vectors is given by Formula 2.4:

$$s\left(\vec{q},\vec{d}\right)=\frac{\vec{q}\bullet\vec{d}}{\left|\vec{q}\right|\times\left|\vec{d}\right|}=\frac{\sum_{i=1}^{n} w(t_i,q)\times w(t_i,d)}{\sqrt{\sum_{i=1}^{n} w(t_i,q)^2}\times\sqrt{\sum_{i=1}^{n} w(t_i,d)^2}} \tag{2.4}$$

In Formula 2.4, $w(t_i,q)$ is the *tf-idf* of the term t_i in the query q and $w(t_i,d)$ is the *tf-idf* of the term t_i in the document *d*. The similarity between the two vectors allows obtaining a ranked list of documents according their relevance to the query. The documents that obtained a higher value of similarity are in the top positions of the ranking.

Probabilistic Model

The Probabilistic model, also known as the binary independence retrieval model, is based on the probability ranking principle. A full description was published by Robertson (1977), although the model was first introduced by Robertson and Sparck Jones (1976). The main objective of the model is to find the *ideal answer set*, by using a probabilistic framework [Baeza-Yates and Ribeiro-Neto, 1999]. In other words, the model tries to describe the inherent uncertainty in the information retrieval process by using the probability theory.

Indeed, the *ideal answer set* is the result list of a query only composed of relevant documents, which are ranked by descending order of probability of relevance. The non-relevant documents cannot be in the result list. Thus, the assumption considered is that relevance is a binary property, that is, a document is relevant or non-relevant. The similarity function is used by this model to divide the document collection into two documents sets for a given query: relevant documents and non-relevant documents. Thus, for each document *d* of the collection, the similarity value between the document *d* and the query *q* is computed using the odds ratio formula as:

$$sim(d,q) = \frac{P(R \mid d)}{P(\overline{R} \mid d)} \tag{2.5}$$

In Formula 2.5, *R* is the set of relevant documents and $\overline{R}$ is the set of non-relevant documents to the given query *q*. Applying Bayes' theorem, the formula is extended as:

$$\begin{aligned} sim(d,q) &= \frac{P(d \mid R) \times P(R)}{P(d \mid \overline{R}) \times P(\overline{R})} \\ &\approx \frac{P(d \mid R)}{P(d \mid \overline{R})} \end{aligned} \tag{2.6}$$

In Formula 2.6, $P(d \mid R)$ and $P(d \mid \overline{R})$ are the probabilities of randomly selecting a document *d* from the set of relevant documents *R* and the complementary set of non-relevant documents $\overline{R}$ to a query *q*, respectively. $P(R)$ and $P(\overline{R})$ are ignored; seeing that they are the prior probabilities of a relevant document and a non-relevant document, the probability is the same for all the documents in the collection.

Such as the Boolean model, the Probabilistic model assumes the absence of term relevance weighting, since it assigns a binary weighting to each term *t* of a document *d*, which is used to define its presence or absence in that document. By using the terms independence assumption, the similarity can be written as Formula 2.7.

$$sim(d,q) \approx \frac{\prod_{i=1}^{n} P(w_i \mid R)}{\prod_{i=1}^{n} P(w_i \mid \overline{R})} \tag{2.7}$$

In Formula 2.7, $P(w_i | R)$ is the occurrence probability of a term w in the relevant documents set of the query q and $P(w_i \mid \overline{R})$ is the probability that the term w occurs in the non-relevant documents set of the same query.

The Probabilistic model became very hard to implement due to the complexity of dividing all documents of the collection in the two sets: relevant documents and non-relevant documents. The relevant documents set is not known at the very beginning, i.e., immediately after the submission of the user's request, since there are no retrieved documents. Thus, it is necessary to estimate a value for the prior probabilities of a term w for computing $P(w_i | R)$ and $P(w_i \mid \overline{R})$. Some methods for such estimation are proposed [Manning et al., 2008]. After that, the probabilities are recalculated by performing iterative retrieval.

Divergence From Randomness Model

Amati and Van Rijsbergen (2002) proposed a new type of probabilistic weighting model for information retrieval based on the concept of Divergence From Randomness (DFR). The idea underlying this concept is that the information carried by a word in a document, namely the informative content of a word, is stronger when the frequencies of a term within a document and within the document collection show a great divergence. In other words, an informative word of a document is a word with few occurrences in the collection, but with many occurrences in the document. So, the hypothesis defined by these models is that the informative content of a word can be measured by analyzing the divergence of its frequency distribution versus the frequency obtained under of a randomness process that is a distribution of non-informative words.

The weight of a word in a document is given by a probability, such as language models. However, these models follow a non-parametric approach, estimating this value by using a combination of different probability distributions. On the contrary, language models need a data driven methodology to compute the probability.

Amati and Van Rijsbergen (2002) created a DFR framework that has originated a generation of weighting models based on a combination of the following elements: a randomness model, an information gain model and a term frequency normalization model. The randomness model estimates the informative content of a word relative to all collection. The information gain model gives the informative content relative to a subset of the collection. This subset, named elite set of the word, is defined as the set of the documents that contain the word. The term frequency normalization model provides the normalization of the document size. It first computes the average size of the document, then, recalculates the frequency of the term considering the normalized document size.

The DFR framework gives the possibility of choosing different probability distributions based on urn models to define the basic randomness models. Examples of the used ones are the binomial distribution, the Poisson distribution, the Bose-Einstein statistics, the inverse document frequency model and a mixed model using Poisson and inverse document frequency. The idea underlying an urn model is to determine the probability of drawing one ball from an urn with a set of balls with different properties, by pure chance. The weight of a term in a document *w(t|d)* is a decreasing function of two probabilities, given by the product of the two informative content functions:

$$w(t \mid d) = Inf_1(t \mid D) \times Inf_2(t \mid E) \qquad (2.8)$$

The function Inf_1 is the informative content of a term *t* in whole collection *D* and Inf_2 is the informative content of a term *t* in the elite set *E*. Inf_1 is a decreasing function of probability P_1 that gives the probability of having *tf* occurrences of a term *t* in a document *d*, by pure chance, taking into account the randomness model applied. The DFR models assume that the non-informative words are randomly distributed on all documents of the collection. In contrast, the informative words obtain a little probability according to the randomness model, since they do not follow that behavior. This assumption is similar to the information retrieval models based on the inverse document frequency measure *idf*. Thus, Inf_1 is given by:

$$Inf_1(t \mid D) = -\log_2 P_1 \qquad (2.9)$$

The information gain model estimates Inf_2 using a probability function P_2 that gives the probability of a term *t* being informative for document *d* of the elite set *E*, considering this occurrence accidental. A term *t* with many occurrences in a document has a very high probability to be a descriptor for that document. So, the risk of accepting this term has an informative term of the document is minimal. In other words, if P_2 is low for the term *t* within document *d*, then the informative content of the term is high and is given by Formula 2.10:

$$Inf_2(t \mid E) = 1 - P_2 \qquad (2.10)$$

Thus, the formula to compute the weight of the term *t* in a document *d*, considering the term frequency, is given by:

$$w(t \mid d) = (-\log_2 P_1) \times (1 - P_2) \qquad (2.11)$$

Formula 2.11 is a function of four random variables *w(F, tf, n, N)*, where:

- *F* is the total number of occurrences of term *t* in its elite set *E*;
- *tf* is the frequency of the term *t* in the document *d*;
- *n* is the size of the elite set *E*;
- *N* is the size of the collection.

As the term frequency *tf* depends on the size of the document, the calculation of Inf_2 requires the recalculation of the term frequency in relation to the document size, by using a term frequency normalization model. The normalized term frequency *tfn* is given by Formula 2.12.

$$tfn = tf \times \log_2 (1 + c \times \frac{avg\_s}{s}) \qquad (2.12)$$

In Formula 2.12, *c* is a tunable parameter, *s* is the size of the document and *avg_s* is the average size of the collection. Then, the normalized term frequency *tfn* is used to compute the two informative content Inf_1 and Inf_2.

In fact, considering the correct term frequency – *tfn*, Formula 2.11 is a function of six random variables *w(t|d)=w(F, tf, n, N, avg_s, s)=w(F, tfn, n, N).*

The DFR models assume the following matching function of the relevant documents *d* for a query *q*:

$$w(d \mid q) = \sum_{t \in q} tfq \times w(t \mid d) \qquad (2.13)$$

2.3 Temporal Information Retrieval

Temporal information retrieval is an emerging research topic in the field of information retrieval that takes advantage of the time dimension existent in documents and queries to improve the effectiveness of retrieval. A wide range of retrieval tasks can benefit by incorporating the temporal information [Alonso et al., 2011, Campos et al., 2014a]. This section shows some examples of retrieval tasks, such as dating documents [de Jong et al., 2005, Kanhabua and Nørvåg, 2008, Nunes et al., 2007], time-aware ranking [Yu et al., 2004, Li and Croft, 2003, Jones and Diaz, 2007, Dakka et al., 2008], future search [Baeza-Yates, 2005], text search over temporally versioned documents [Berberich et al., 2007, Lan et al., 2014], query expansion [Amodeo et al., 2011, Radinsky et al., 2011, Whiting et al., 2011], and Web crawling [Pereira, 2013, Pereira et al., 2014].

Temporal information has been explored by information retrieval models in various ways. These models can be classified into two categories with regard to the essence of the temporal information used: one, *metadata-based analysis*, takes into account only the document timestamp, and the other one, *content-based analysis,* uses temporal references embedded in the content of documents and queries. Our work fits into this latter category, although to the best of our knowledge, none of the models in this category takes advantage of the relationship between dates and words of the documents, as we propose in our work.

Temporal information retrieval has also some approaches, which exploit the link structures of Web pages. Yu et al. (2004) modified the Page Rank algorithm [Brin and Page, 1998], incorporating the time dimension. The date of citation is used in the ranking of documents to improve the quality of the search results. Dai and Davidson (2010) proposed a link-based ranking method considering the freshness of the page content, both the original Web page and the pages linked to it.

Some interesting approaches follow the *metadata-based analysis*, considering only the document timestamp. This does not avoid obtaining very interesting results related to time. Li and Croft (2003) proposed an approach focus on *recency queries*, but temporal information is not explicitly referred in the information needs. The creation date of documents is used to promote the more recent documents of each query, giving them a higher score in the probability of relevance. This approach applied time into the statistical language models proposed by Lavrenko and Croft (2001), namely time-based language models.

There is also research based on the establishment of a temporal profile for the queries, considering the documents timestamp. Diaz and Jones (2004) proposed an interesting approach that creates a temporal profile of a given query using the distribution of the publication date of the retrieved documents, in order to determine the period of time relevant to the query. This approach takes into account the higher weight to the time unit considered, for instance, days, where there are more occurrences of the query terms. With this approach, Jones and Diaz (2007) classified queries into three temporal classes: atemporal, temporally unambiguous and temporally ambiguous. Based on this work, Rode and Hiemstra (2006) created mechanisms for relevance feedback and query disambiguation and clarification. Their approach uses the query profiles to reformulate the original query, detecting and providing visualization of query ambiguity. A summary report about the expected query is displayed to the user in order to help him/her in the query clarification.

Dakka et al. (2008) observed that the publication time of the documents in a news archive is an important aspect to improve the results when it is used with the topic similarity in the ranking of documents. They only work with a particular class of queries, which they called time-sensitive queries, although these queries can also be considered temporal queries.

Berberich and Bedathur (2013) focused their recent work on ambiguous queries. They introduced the notion of temporal diversity by using the publication date of documents. Their method determines the relevant documents published at diverse times to a query.

Sato et al. (2003) also defined a temporal query, determining the time point or interval where the document exists. This information is used to rank the results, taking into account the time metadata of documents to obtain the fresh information retrieval. Another approach to post-process the search engine results was proposed by Jatowt et al. (2005). The algorithm ranks the documents by their freshness and relevance, which are estimated by analyzing changed contents between a current version and archived versions of Web pages. In summary, documents in top were significantly and recently modified.

Another interesting approach was proposed by Berberich et al. (2007) to search over temporally versioned documents, considering a specified time. This approach uses an inverted index with explicit temporal information. More precisely, the time interval in which the payload information was valid is incorporated in postings. To avoid an index size explosion, since an entry index is created per term per document version, some techniques to improve performance are also considered in this approach.

Working also with versioned documents, Lan et al. (2014) proposed the Interval Window based Algorithm (IWA) to process continuous temporal top-*k* query. Its main objective is to find documents that are frequently in the top-k of the retrieved documents for a given query, considering the query specified time and the weights of different time intervals.

The proposals described next do not use the content of documents, nor the document timestamp, but have a similar objective, which is to incorporate the time dimension in models. Metzler et al. (2009) created a model that favors the implicit temporal intent of the user, using the query log with frequency information to obtain the implicit year of the queries. Shokouhi (2011) proposed an approach to detect seasonal queries by using time-series decomposition techniques. A time-series is generated for each query, where each data point represents the query frequency for that point in time.

Efron et al. (2012) proposed an approach to improve information retrieval systems when used in collections of short texts. This approach is based on document expansion, assuming that the short texts tend to be a single topic. Short documents are submitted as pseudo-queries. A temporal profile is created for each pseudo-query with the publication date of documents in the result set. These profiles are helpful when latter used during time-sensitive document ranking.

It is also interesting to mention the work based on *content-based analysis*, i.e., the temporal relevance of the document content, which can be used for a diversity of different applications, such as searching the future [Baeza-Yates, 2005], clustering documents or exploring search results using timescales results [Alonso et al., 2009b], and also generating temporal snippets for search [Alonso et al., 2009a].

There are also attempts to use the uncertainty present in some temporal expressions that are vague or incomplete, introducing fuzzy representations of temporal information that is implicit in the user's mind [Kalczynski and Chou, 2005]. Specific temporal expressions with an inherent uncertainty such as *"in the 1990s"*, were integrated into a language retrieval model by Berberich et al. (2010).

Recently, Brucato and Montesi (2014) proposed a temporal ranking model based on metric spaces. Temporal expressions are mapped into time intervals that are used to model the temporal scope of documents and queries. The similarity measure of documents and queries is defined by combining the textual and temporal similarities.

Temporal information in the documents content is also used for processing of time-sensitive queries with explicit or implicit temporal information. The study of these queries has also been the focus of the recent research [Kanhabua and Nørvåg, 2012, Lin et al., 2014, Campos et al., 2014b, Gupta and Berberich, 2014, Gupta and Berberich, 2015]. Our work is also focused in these queries, but it takes into consideration the temporal relationships between words, unlike the other works.

Another interesting application of the temporal information is obtaining the document timestamp, i.e., determining the time of a document. Some statistical language models also estimate the time of documents, using the temporal information of the document content [de Jong et al., 2005, Kanhabua and Nørvåg, 2008].

Timestamping of Web pages is another approach also based on the concept of dating documents duly adapted to the Web context. To determine the Web page publication or modification time, it is requires not only the content of the non-timestamped Web page, but also the documents containing links to this Web page. Nunes et al. (2007) proposed an approach based on the neighbors of the non-timestamped document that were classified as: incoming, which are documents having a link to the non-timestamped document, outgoing, which are documents pointed to by the non-timestamped document, and assets, which are all URLs referenced in non-timestamped document.

The estimate time for the non-timestamped document is computed by the average of Last-Modified dates extracted from neighbor documents.

Temporal information is also used to improve automatic query expansion. Although none of the approaches is based on *content-based analysis*, since they only consider the timestamp of documents, they are the closest to our work.

Keikha et al. (2011a, 2011b) proposed a query expansion approach for blogosphere, which takes into account the publication date of blogs and queries to select the terms for expansion. In this approach, the blogs and queries are represented in a timeline with the granularity of days. The selection of the top-n terms for each day, considering the time space of a query, is carried out with the KL-divergence between the term distribution of the day and the whole collection. This approach also identifies the dynamics of topics both in aspects and vocabulary usage over time.

There are already a number of research papers that establish temporal correlations between different words or concepts [Amodeo et al., 2011, Radinsky et al., 2011, Whiting et al., 2011]. These studies are concerned with the temporal profile correlation of words based on the date of publication of the documents.

Radinsky et al. (2011) proposed an approach to compute the degree of semantic relatedness of words, creating a temporal concept dynamics. They assume that *"concepts that have similarly over time, are semantically related"*. First, each word is represented with a weighted vector of concepts. Then, a time series is created by each concept, considering that a concept is a sequence of words and quantifying concept occurrence for each period of time. Finally, it scales each time series according to the concept original weight represented in vectors. The semantic relatedness of two words is computed by comparing the two vectors.

Amodeo et al. (2011) promoted the effectiveness of the retrieval in a blog search domain by the correlation between time and relevance using the publication date of documents. The approach assumes that the relevance of a topic is related to the amount of documents published in a given period of time. A spike in the publication of documents related to the topic shows an increasing interest for the topic. Thus, starting from the retrieved documents set of a query, Amodeo et al. (2011) built the time series with the publication date of documents. Then, these time series were compared with the one built for the true relevant documents. The conclusion of this study is that spikes in the time series of the two documents set are related. Thus, it was assumed that pseudo-relevant documents, i.e., the top-k most relevant documents used by the query expansion algorithm are the documents in the highest peaks. Documents outside from the peaks are considered as not relevant.

Whiting et al. (2011) determined the importance of the using long-term temporal profiles in improving of the term selection process of automatic query expansion, considering the temporal profile correlation between terms from the pseudo-relevant documents and temporally significant terms. Long-term temporal profiles of terms are obtained from the documents timestamp of a large collection. The query terms with the highest kurtosis are selected to temporally correlate with the terms found in pseudo-relevant documents.

2.4 Temporal Retrieval Preprocessing

Documents and queries must be processed in order to extract the temporal information found in texts. This information is put into a normalized format in order to be included in the information retrieval model. Traditionally, this is the only information required by retrieval models, since the time is introduced in these models without any dependence with the words, contrary to our work. Our approach is distinguished from previous ones by the use of the temporality of words. This means that it is possible to know in which date (or dates) a given document word is referred to. The temporal relationships between words are obtained by the text segmentation based on time discontinuities found in the text. Each segment is composed of a set of sentences that share the same temporality. In this way, all the words of a given segment are temporally related.

In our work, the temporal retrieval preprocessing consists of extracting the temporal information from documents in a first step, and then segments the texts according to the temporal discontinuities identified.

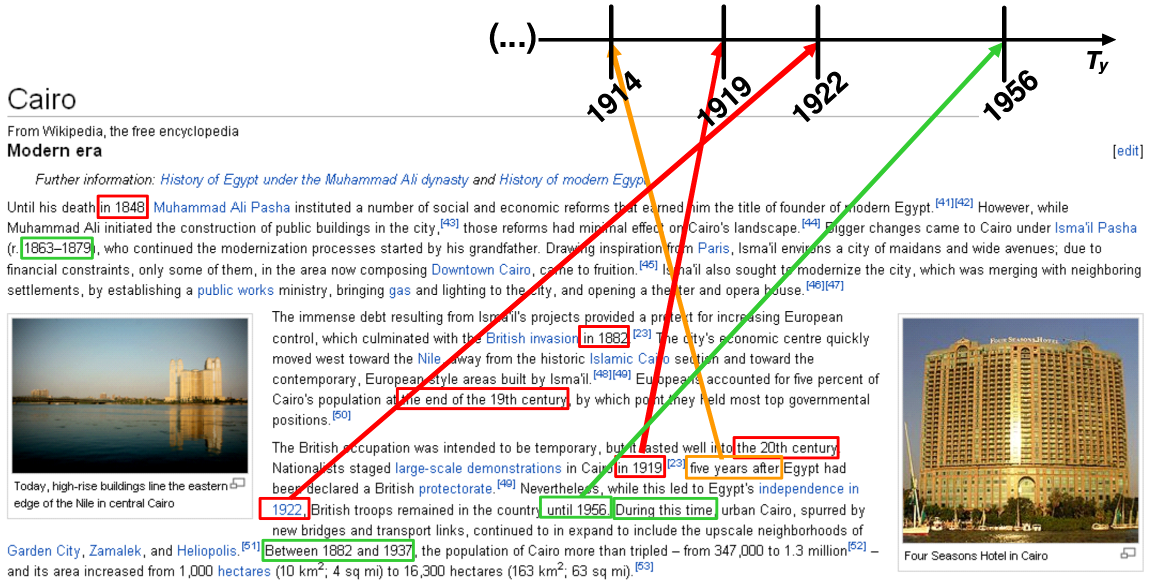
Since there is no tool to process documents in the Portuguese language for the purposes, we developed a system from scratch to carry out the task of temporal retrieval preprocessing. The testbed system is composed of two tools. Starting from Portuguese texts, one of the tools, first, identifies and classifies the temporal expressions; and then, it put these expressions in a pre-defined format. The other tool segments the text, using the normalized temporal expressions. The proposed approaches for the task of temporal retrieval preprocessing, and the tools developed following the approaches are detailed in Chapter 3.

2.4.1 Information Extraction

Temporal information can be found in metadata of documents and queries, such as the creation or publication date of documents, or the submission date of the query. Besides that, temporal information can also be found in the text of the documents and the queries. Indeed, the extraction of this information is not a trivial task.

Temporal information extraction is responsible for recognizing the temporal expressions, which means to identify and classify the expressions expressed in texts by different ways, such as represented in Figure 2.3. After that, the resolution of these expressions is also carried out. The main objective is to map the temporal references found in the document content into a timeline, also according to the way that these references are expressed in the text.

Figure 2.3 shows an example of the temporal information extraction in the Cairo Wikipedia page (retrieved November 20th, 2010). The temporal information is expressed in texts by different ways, such as *"in 1848"*, *"1863-1879"*, *"in 1882"*, *"the end of the 19th century"*, *"20th century"*, *"in 1919"*, *"five years after"*, *"in 1922"*, *"until 1956"*, *"during this time"*, *"between 1882 and 1937"*. However, the main objective is to convert that information into a normalized and machine-readable format, but not all of them can have a normalized format. In this example, the information is processed to be represented in a timeline defined in granularity by *year*. Therefore, the expression *"the end of the 19th century"* cannot be normalized in this timeline. If the timeline granularity is century, it refers to 19th. However, some vagueness is underlying in this expression when the granularity is year. It is not possible to precisely determine the year(s) referenced by the expression.



Cairo

From Wikipedia, the free encyclopedia

Modern era [edit]

Further information: History of Egypt under the Muhammad Ali dynasty and History of modern Egypt

Until his death in 1848 Muhammad Ali Pasha instituted a number of social and economic reforms that earned him the title of founder of modern Egypt.[41][42] However, while Muhammad Ali initiated the construction of public buildings in the city,[43] those reforms had minimal effect on Cairo's landscape.[44] Bigger changes came to Cairo under Isma'il Pasha (r. 1863–1879), who continued the modernization processes started by his grandfather. Drawing inspiration from Paris, Isma'il envisions a city of maidans and wide avenues; due to financial constraints, only some of them, in the area now composing Downtown Cairo, came to fruition.[45] Isma'il also sought to modernize the city, which was merging with neighboring settlements, by establishing a public works ministry, bringing gas and lighting to the city, and opening a theater and opera house.[46][47]

Today, high-rise buildings line the eastern edge of the Nile in central Cairo

The immense debt resulting from Isma'il's projects provided a pretext for increasing European control, which culminated with the British invasion in 1882.[23] The city's economic centre quickly moved west toward the Nile, away from the historic Islamic Cairo section and toward the contemporary, European-style areas built by Isma'il.[48][49] Europeans accounted for five percent of Cairo's population at the end of the 19th century, by which point they held most top governmental positions.[50]

The British occupation was intended to be temporary, but it lasted well into the 20th century. Nationalists staged large-scale demonstrations in Cairo in 1919,[23] five years after Egypt had been declared a British protectorate.[49] Nevertheless, while this led to Egypt's independence in 1922, British troops remained in the country until 1956. During this time, urban Cairo, spurred by new bridges and transport links, continued to in expand to include the upscale neighborhoods of Garden City, Zamalek, and Heliopolis.[51] Between 1882 and 1937, the population of Cairo more than tripled – from 347,000 to 1.3 million[52] – and its area increased from 1,000 hectares (10 km^2; 4 sq mi) to 16,300 hectares (163 km^2; 63 sq mi).[53]

Four Seasons Hotel in Cairo

Figure 2.3: Timeline for Cairo Wikipedia page.

The normalized format of a temporal expression is a discrete representation of time, denoted *chronon* by Alonso et al. (2009). It is determined according to the classification of temporal expressions, such as date, time, interval, duration, and frequency. The document timestamp or a date in the neighborhood of the recognized expression can be useful to inference a new date, allowing an incomplete or implicit date to be mapped into a more specified date. For example, in Figure 2.3 *"five years after"* which is not an explicit temporal expression, was mapped to *1914* using the near date *1919*.

Although a plethora of works exists for the area of temporal information extraction in English texts, to the best of our knowledge, none of them automatically creates a set of expression patterns and applies it for temporal expressions recognition in Portuguese texts, such as our proposal approach.

In our approach, the matching of expression patterns requires a sentence-by-sentence processing and does not require a linguistic analysis of the text. However, temporal information extraction systems are mainly based on term-by-term processing, by using term linguistic characteristics for temporal expressions identification, such as techniques presented by Mani (2003). Mani and Wilson (2000) proposed an annotation scheme to represent dates and time in temporal expressions, which is based on a diverse set of manually defined and automatically discovered rules. This approach uses finite-state automata, focusing on core expressions and neglecting prepositions. An alternative approach proposed by Schilder and Habel (2001) introduced prepositions on the finite-state automata. This approach is based on the work done by Allen (1983) in the detection of temporal intervals. The approach proposed by Makkonen and Ahonen-Myka (2003) employs functional dependency parsers and finite-state automata for temporal expressions recognizing, based on the clustering of the terms occurring in the temporal expressions into categories, such as *baseterms*, indexical, temporal, etc.

Vazov (2001) described a very different approach for the temporal expressions identification in French documents, based on a Context-Scanning Strategy (CSS). Although this is a rule-based approach, temporal expressions are not only determined by world knowledge, but they are configurations integrating linguistic data found in the text. The CSS is carried out in two steps. The first step is to identify temporal markers in text. If the temporal marker is an autonomous temporal expression, for example *daily*, the process finishes. Otherwise, the system launches the second step. A left-to-right (and right-to-left) chart-parser is used to identify larger temporal expression, for example *two weeks before Carnival*. The objective of the second step is to determine which lexical units near the marker *before* belong to the temporal expression. This approach does not resolve temporal expressions. Although the recognition task also uses regular expressions as our approach, the CSS uses term linguistic characteristics, unlike our approach.

There are also several strategies to resolve temporal expressions [Mani and Wilson, 2000], [Ahn et al., 2005], [Schilder and Habel, 2001] or to assign timestamps to event clauses [Filatova and Hovy, 2001].

TimeML (Time Markup Language) [Pustejovsky et al., 2005, Saurí and Pustejovsky, 2009] is a specification language for annotation of events and temporal expressions in natural language text. TimeML takes into account the grammatical features such as verb tense and temporal aspects found in the text to provide the annotation. TimeML addresses issues of time stamping (events and temporal expressions), ordering events, reasoning about the persistence of events and resolution of temporal expressions.

TIMEX3 is the structured data specified in TimeML to mark up the temporal expressions [Saurí et al., 2006]. It was built from earlier annotation schemes of temporal expressions, TIMEX [Setzer, 2001] and TIMEX2 [Ferro, 2001].

There are also some tools available, for example, the TARSQUI toolkit [Verhagen and Pustejovsky, 2008] and HeidelTime [Strötgen and Gertz, 2013]. Unfortunately, the annotation schemes or tools were not directly applicable to the Portuguese language until a short time ago. Costa and Branco (2012) proposed an adaptation of the TIMEX3 tag defined in TimeML for Portuguese. Recently, this was included in the temporal tagger HeidelTime [Strötgen and Gertz, 2015].

Since the extraction of temporal information is a task that depends on the language, the systems built for English language processing cannot be directly applied. Although there is already some work done in the processing of the Portuguese language, which can be also important in the direction of the temporal information extraction, the area of temporal information extraction in Portuguese texts is still not the focus of a significant amount of work.

PALAVRAS, for instance, is an automatic grammar and lexicon-based parser for unrestricted Portuguese text [Bick, 2000]. This system is an important tool for Portuguese text annotation, even though it uses a generic approach to handle temporal expressions. Baptista (2003a, 2003b) worked with lexical finite-state transducers to describe and mark up complex multiword temporal adverbs in Portuguese text, just involving the time-related noun *ano* (year). Móia (2006) presented a study focused only on the duration expressions in the Portuguese language, by using English language for comparison purposes. Móia (2001) also proposed an interesting semantic characterization of time-denoting expressions and temporal locating adverbials, which is useful in the process of temporal expressions disambiguation.

To the best of our knowledge, there are only two temporal information extraction tools developed to process Portuguese texts. The XTM (XIP Temporal Module) tool developed by Hagège et al. (2010), and Hagège and Tannier (2008), to extract temporal information from Portuguese texts, among other languages, such as, English and French. The XTM is an extension of the syntactic analyzer XIP that provides a deep analysis approach of texts. This tool is based on rules and word-by-word processing. Unfortunately, it is not a public domain available tool. More recently, Costa and Branco (2012) developed the LX-TimeAnalyzer to annotate temporal expressions in Portuguese texts analyzing word-by-word and following the TIMEX3 specifications of annotation.

Unlike the works mentioned above, our proposal for extraction of temporal information does not require a linguistic analysis of the text, using a sentence-by-sentence processing. In addition, it uses lexical patterns generated from Portuguese texts.

2.4.2 Text Segmentation

Text segmentation is the process of dividing the text into smaller units, which can be of different types and sizes, such as, morphemes, clauses, sentences, paragraphs, topics, and so on. In fact, the granularity and unit types depend on the purpose of the application to the segmentation.

The objective of the segmentation by topic is to identify the boundaries of topic changes in text streams and divide the texts into a set of contiguous sentences or paragraphs sharing the same topic, called text segments [Hearst, 1994, Beeferman et al., 1999, Misra et al., 2009]. This way of partition of the text has become an important technique over the last few years for several applications of text processing, such as information retrieval, passage retrieval, text mining, text summarization and discourse analysis [Hearst, 1997, Ji and Zha, 2003, Misra et al., 2011].

Our segmentation approach also intends to identify boundaries, but in this case, given by the temporal discontinuities found in the text. All the words of a segment share the same temporality.

In some text collections, there is not an explicit notion of document, as for example the streams of transcripts carried out by automatic speech recognition systems. Usually, these text streams have distinct topics without any marked boundaries between them. Topic segmentation can be crucial to make these text collections readable and understandable for other systems. Text summarization is another application, where the topic segmentation can be used to select segments of text containing the main ideas for the summary requested. In information retrieval and passage retrieval, the topic segmentation can provide as a result only some relevant parts of documents, which are closer to the user's need, instead of a set of documents, since many times the users can be satisfied by the presentation of only a set of text segments or passages.

Traditional information retrieval systems only use topical segmentations in which text is divided into multiple segments thematically coherent and distinct. The research literature is abundant on this subject [Ponte and Croft, 1997, Hearst, 1997, Ji and Zha, 2003, Labadié and Prince, 2008, Misra et al., 2009]. However, the literature is very limited about temporal text segmentation. To the best of our knowledge, there is no research work focused on the temporal segmentation of Portuguese texts.

Although the work by Ji and Zha (2003) is not focused on temporal text segmentation, it follows an approach very different from the traditional approaches of topic segmentation. This work is very interesting because it is domain-independent and based on image segmentation techniques. In this approach, a sentence distance matrix is used in order to obtain the sentence cohesion information in a document, by applying the anisotropic diffusion technique proposed by Perona and Malik (1990). Each sentence is represented by word-frequency vectors. The matrix is formed by computing all pairwise distances of the sentences in the document. The topic boundaries are obtained by applying dynamic programming to the sentence cohesion information.

The work of Jean-Louis et al. (2010) is closed to ours, since the extracted temporal information is also used as a basis for segmentation, though the aim is different. The text is segmented into pieces that correspond to a single event. Based on statistical machine learning methods, it claims very good results in the domain of seismic events when compared to a heuristic approach developed by experts.

Although Brawsen et al. (2006) also defined a method for temporal segmentation of texts, the purpose is very different, which is the understanding of the temporal flow of discourses in a specific context – clinical narratives. This method, which gives more primacy to precedence relation between the segments, relies on the construction of a directed acyclic graph that provides a chronological order of the segments; indeed, this can be considered as a coarse annotation scheme, an alternative of others annotation scheme, such as TimeML.

Other interesting approach of temporal segmentation is applied to topics collected from a text collection, instead of text. Pan et al. (2013) are building a visual tool to help users in analyzing the temporal evolution of topics. First, this tool automatically obtains the topics of a text collection, and then, it breaks each of those topics into sub-topics represented in a timeline. The timestamp of the relevant documents to a topic is used to determine the temporal locality of the sub-topic, which is the time interval where these documents are predominant.

Regarding the related work, although our approach takes into account the temporal discontinuities found in text, it is distinguished by the fact that we want to divide text into temporally coherent units. In fact, the unit is a set of sentences tagged with the same timestamps that are the *chronons* found in that piece of text. We are not concerned with the chronological order of the segments, but with the relationship between segments and accurate dates, in order to later use this information in retrieval systems. Our algorithm is based on simple heuristic rules, though most of the existing segmentation tools are based on supervised machine learning algorithms. Our option is justified by the lack of suitable Portuguese training collections, since usually these algorithms do not achieve good results without suitable training collections.

2.5 Summary

This chapter begins by explaining how time can be important in information retrieval systems, since it can play an important role for text understanding. Exploring this idea, our approach is based on an assumption that words can be temporally related, taking advantage of this relationship to improve the retrieval effectiveness. Words with same temporality can share the description of the facts and events also temporally related. Our approach has already been applied to retrieval systems, presenting auspicious results. In addition, a temporal focused Web crawler was also developed based on our approach.

A temporal retrieval preprocessing must be performed in documents and queries to incorporate the time dimension in retrieval systems. Besides the extraction of temporal information, our approach needs also to temporally segment texts in order to obtain the temporality of words. Thus, in this chapter, we present the two topics underlying this temporal preprocessing, which are temporal information extraction and temporal text segmentation.

This chapter addresses different contexts in which temporal information plays an important role to achieve better results. A brief description of the related works carried out in the important areas of research, namely, information extraction, text segmentation and temporal information retrieval, is also presented.

In addition, this chapter gives the description of the fundamental concepts and techniques of information retrieval, which are useful for understanding our contributions presented in the following chapters.

Chapter 3

Temporal Extraction and Segmentation

To obtain temporal information in a format that can be incorporated in retrieval models, a temporal preprocessing must be done in documents and queries. This chapter describes our approach to obtain this information in Portuguese texts. The developed tools are duly explained in this chapter. In addition, our approach needs to segment the texts. The goal is to find temporal discontinuities in the content of annotated documents to split the text into coherent segments tagged with timestamps. We also developed a tool for this purpose, which is also presented in this chapter. The chapter also provides the evaluation carried out and the discussion of the results.

3.1 Introduction

Following the assumption that the temporality of words can improve the effectiveness of retrieval systems, our approach requires the establishment of temporal relationships between words, which are obtained by the text segmentation based on the discontinuities found in the text. Therefore, words in the same segment of the documents share the same temporal references.

Since there is no tool to process documents in the Portuguese language for this purpose, we propose two approaches to obtain the temporality of words. One approach is to extract the temporal information from texts, putting it into a normalized format in order to be used by the segmentation. The second approach is to temporally segments establishing the temporal relationships between words. This approach requires the information given by the text extraction.

Based on the two proposed approaches presented in Sections 3.2 and 3.3, we have developed a system from scratch to carry out the required temporal retrieval preprocessing composed of the *Extraction* tool, and the *Segmentation* tool, respectively. The tools were developed in Perl language. Figure 3.1 shows the interconnection of the modules of the two tools.

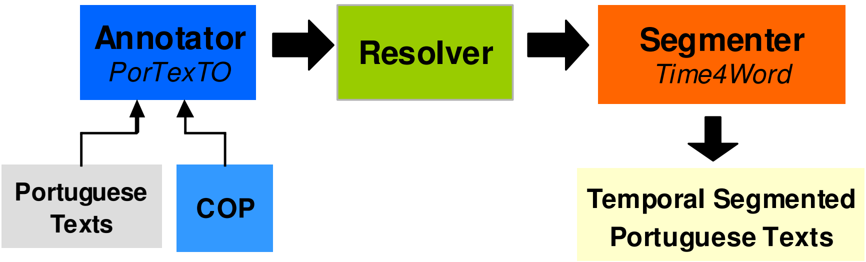


Figure 3.1: Testbed system for temporal retrieval preprocessing.

The *Extraction* tool is composed of three modules: Co-Occurrence Processor (henceforth COP), Annotator, named PorTexTO (PORtuguese Temporal EXpressions TOol) and Resolver. The Segmenter module, name *Time4Word*, is the single of the *Segmentation* tool. Note that in this work, all the modules are used together; however, each one can be used individually.

Starting from Portuguese texts, the Annotator module identifies and classifies the temporal expressions; and then, the Resolver module put these expressions in a pre-defined format. As result, the extraction tool gives the original text marked up with the classification and the normalized format of the temporal expressions identified. The identification of the temporal expressions is performed using a set of temporal expressions patterns, obtained by the COP module.

The Segmenter module uses the annotated texts obtained by the extraction tool to partition the text into temporally coherent segments, providing the relationship between words and time found in each document.

In the development of this system, we tried to find a trade-off between efficiency and simplicity, while maintaining an appropriate level of effectiveness. Indeed, it is not important to recognize all the temporal expressions of documents, since some of them cannot be put into a normalized format in order to be used by the segmentation, as for example, generic expressions such as *"within some days"*.

The evaluation focused on the effectiveness of the extraction and segmentation approaches. Indeed, each approach was independently evaluated.

The toolset was built to identify temporal segments in order to determine the words temporality that is introduced in retrieval model. This way, it was allowed the evaluation of our proposal of query expansion methods, described in Chapter 5. In addition to the originally intended utilization, the toolset was also used to build a Web crawler [Pereira, 2013]. The Web pages found by the crawler were segmented. The temporal segments were used to decide which links of the Web page segmented will be visited by the crawler. Section 5.4 presents a brief description of the temporal focused Web crawler. In fact, the toolset can be used in several contexts.

In the following subsections, we describe the proposed approaches for temporal retrieval preprocessing, starting with the temporal information extraction and then, presenting the temporal segmentation.

3.2 Temporal Information Extraction

The extraction of temporal information is a complex task. Temporal information is expressed in texts by different ways, and even for a trained human reader it would be difficult to identify all temporal expressions. Besides that, the information must be converted to a normalized and machine-readable format. In this sense, the information is processed to be represented in a timeline defined in a given granularity, such as *year*, *month*, *day*, etc.

Our extraction approach is focused on Portuguese language processing. In the extraction process, first the recognition of temporal information is carried out, which consists of the identification and semantic classification of temporal expressions. Then, it is performed the resolution of temporal expressions, which means to put the temporal expressions in a normalized format, since these expressions can be expressed in texts by different ways.

In the following, we present the main concepts of the extraction approach; the details are described in Sections 3.2.2 and 3.2.3.

In our approach, the temporal expressions are classified as date, time, interval, duration, and frequency, following the guidelines defined by Hagège et al. (2008a). The guidelines fully describe the classification to be given to the different temporal expressions. A temporal expression is classified as date when the expression can have an entry in the calendar system. For example, the expression *no próximo ano* (next year) is a date; it is 2016 in the calendar system. These guidelines also define the text that must be annotated. For example, in the sentence *"Rio de Janeiro é a cidade anfitriã para os próximos Jogos Olímpicos em 2016"*[2], the temporal expression is *"em 2016"* (in 2016) and is not *2016*, since the preposition must be included in the temporal expressions, according these guidelines. Temporal expressions and their classification are marked in the original text following the annotation scheme TimeML [Pustejovsky et al., 2005, Saurí et al., 2006].

Table 3.1 summarizes these guidelines and presents some examples in Portuguese. The English translation is at the footer of the page.

Classification	Description	Example[3]
Date	entries of the calendar system	*11/03/2012, hoje*
Time	entries of a clock system (granularity lower than day)	*14:30, 2 p.m.*
Interval	two simple time expressions joined by a connector	*entre Abril e Maio*
Duration	a quantity of time	*durante 2 meses*
Frequency	a repetition on the time	*diariamente*

Table 3.1: Temporal expressions classification: some examples.

The annotation scheme uses the XML format and comprises a unique identification, a category denoted as TEMPO (*time*), a type (TIPO) and a subtype (SUBTIPO) only for the type TEMPO_CALEND (*calendar reference*) with the following options:

TIPO=*TEMPO_CALEND* SUBTIPO={*data* (date)*, hora* (time)*, intervalo* (interval)}

TIPO={*duração* (duration)*, frequência* (frequency)}

[2] English version: *Rio de Janeiro is the host city for the next Olympic Games in 2016.*

[3] English version: *11/03/2012, today, 14:30, 2 p.m., from April to May, during 2 months, daily*

The detailed specification of the types and subtypes was presented by Hagège et al. (2008a). Some examples are presented below, where the English translation of the Portuguese sentences is written between brackets.

```
(1) Estive em Berlim <EM ID="1" CATEG="TEMPO"    TIPO="TEMPO_CALEND"
SUBTIPO="DATA">em 2008</EM>.                        (I was in Berlin in 2008)
(2) O jogo começa às <EM ID="2" CATEG="TEMPO"    TIPO="TEMPO_CALEND"
SUBTIPO="HORA">21:00</EM>.                       (The game starts at 9 p.m.)
(3) Estive em França <EM ID="3" CATEG="TEMPO"    TIPO="TEMPO_CALEND"
SUBTIPO="INTERVALO">entre Abril e Maio</EM>.
                                   (I was in France between April and May.)
(4) Visito os meus pais <EM ID="4" CATEG="TEMPO"    TIPO="FREQUENCIA">
todos os dias</EM>.                          (I visit my parents every day)
(5) Eles trabalharam <EM ID="5" CATEG="TEMPO"    TIPO="DURAÇÃO">
durante 2 dias</EM>.                          (They worked during two days)
```

The resolution includes interpretation and normalization of temporal expressions. In the interpretation, the temporal information of the document, such as document timestamp or a date in the neighborhood of the temporal expression, is used to infer a new date. In this way, a recognized expression, which may be incomplete and non-explicit date, is mapped into a more complete specified date. The classification of the expressions and the way that the references are expressed in the text are also of utmost importance.

Temporal expressions can be expressed in different ways: explicit, implicit, relative and vague, according to the proposals defined by Schilder and Habel (2001). Relative/indexical references are expressions that need, at one point in time to be completely resolved and anchored in a calendar/clock system. This means that if this time point was not found, these temporal expressions cannot be resolved. Ahn et al. (2005) defined relative/indexical references as *deictic timexes* or *anaphoric timexes*. Note that vague references are never resolved.

Table 3.2 shows some examples of temporal expressions resolution in the Portuguese language. The English translation is at the footer of the page. Note that it is impossible to resolve the expression *recentemente* (recently). *10-06-2014* is the resolved date of expression *hoje* (today), considering that document timestamp is also *10-06-2014*. The resolved date of *mês seguinte* (the next month) is *07-2014*, considering that a reference to the month of June is in the preceding sentences. The other two references are easily resolved.

References	Description	Example[4]	Resolved date
Explicit	a direct entry in a calendar or clock system	*11/13/2015, 14:30*	*11/13/2015, 14:30*
Implicit	can be directly anchored in the calendar/clock system	*Natal 2014*	*25-11-2014*
Deictic timexes	are resolved by using the document timestamp	*hoje*	*10-06-2014*
Anaphoric timexes	are resolved by using a time point evoked in the text	*mês seguinte*	*07-2014*
Vague	the start and/or end time points are not clear	*recentemente*	*none*

Table 3.2: Temporal references expressed in different ways.

The normalization of temporal expressions consists on the transformation of the dates that are resolved, into a normalized format, anchored in a calendar/clock system by timelines defined by points. This means that resolved dates are mapped into *chronons*.

Figure 3.2 presents some examples of temporal information mapped into *chronons* – normalized dates anchored in a calendar/clock system. This figure shows a Portuguese text "*Eu estou hoje no Cairo, mas chego a Portugal no dia 2 de Dezembro. Vou regessar ao trabalho no dia seguinte a ter chegado.*" (Today, I am in Cairo, but I arrive to Portugal on 2nd of December. I return to work in the day after I arrive.) and the *chronons* extracted from it.

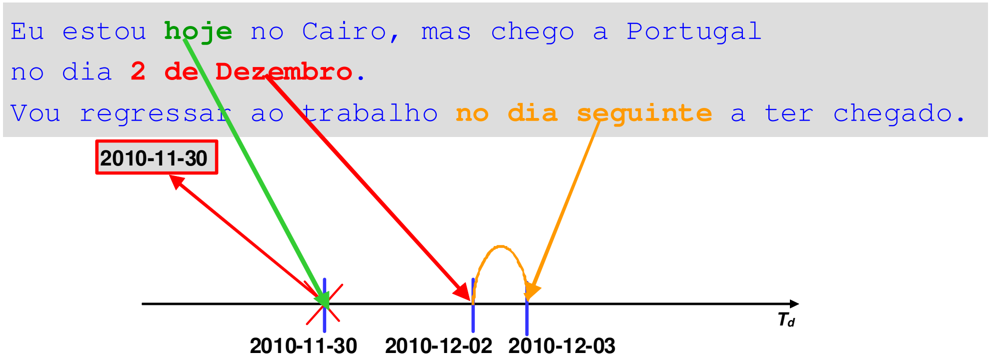

Eu estou hoje no Cairo, mas chego a Portugal
no dia 2 de Dezembro.
Vou regressar ao trabalho no dia seguinte a ter chegado.
2010-11-30
T_d
2010-11-30
2010-12-02
2010-12-03

Figure 3.2: Temporal information extraction of a text in Portuguese.

[4] English version: *11/13/2015, 14:30, Christmas 2014, today, the next month, recently*

In the text, temporal information is expressed in different ways, such as *hoje* (today), *no dia 2 de Dezembro* (on 2nd of December), and *no dia seguinte* (in the day after). The resolved dates are *30-11-2010*, *2-12-2012*, and *3-12-2012*, respectively. The rectangle in red shows the document timestamp (*2010-11-30*) that is required to determine the *chronon* mapped from the expression in green (deictic timexes). The expression in red was used to map the expression in yellow (anaphoric timexes) into a *chronon*.

Since the extraction of temporal information is a very complex task, meaning that covering all the possible cases is difficult to accomplish in practice, our approach focused only on a part of the problem. We try to find a trade-off between efficiency and effectiveness to cover the most suitable cases to be used in retrieval systems. Therefore, less frequently used temporal expressions are ignored. In our approach, the patterns used to identify temporal expressions were derived from the higher probability temporal reference word co-occurrence from Portuguese corpora, namely Web03PT [Martins and Silva, 2004] and HAREM Collection version 2.0, which are available at Linguateca site [lin, nd]. Expressions classified as duration or frequency are annotated but not normalized in this phase of the work.

This section started with the main concepts of the extraction. The rest of the section is organized as follows: Section 3.2.1 presents the system architecture for extraction; Section 3.2.2 details the recognition approach that includes the COP and the Annotator modules. The resolution approach is described in Section 3.2.3. PorTexTO, the Annotator module, participated with a preliminary version in an evaluation contest promoted by Linguateca, a language resource center for Portuguese. The results obtained are fully described in Section 3.2.4. Section 3.2.5 presents the evaluation of the extraction approach.

3.2.1 Temporal Extraction System Architecture

To achieve the objective of enriching the information retrieval indexes with temporal information, it is necessary to process a document in a number of operations, as explained before. The first step is precisely the extraction of the temporal information in Portuguese texts, carried out by a tool developed to this purpose.

Figure 3.3 shows the system architecture of the extraction tool composed of three modules: COP, Annotator and Resolver. In the following, we give a brief description of how they work. The details of these modules are in Sections 3.2.2 and 3.2.3. Appendix A shows some examples of the text processed by these modules.

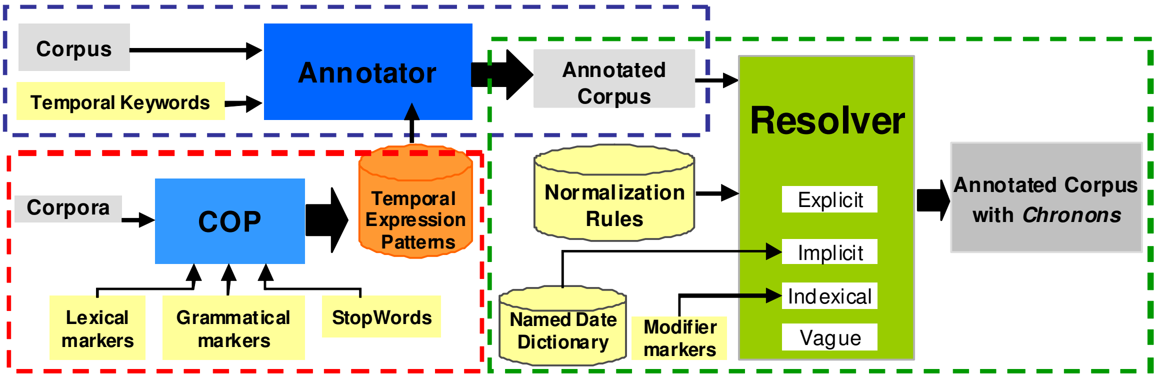


Figure 3.3: System architecture for temporal information extraction.

COP creates the temporal expression patterns that are used in the identification of temporal expressions. It analyzes Portuguese texts, determining the words combination that exists in temporal expressions. The lexical and grammatical markers are used to define the meaningful words of these expressions. For example, the pattern *"No ano passado"* (in the last year) is identified by COP in the processing of the sentence *"No ano passado estive em Berlim."* (In the last year, I was in Berlin.). COP searches in the sentences words that belong to the lexical and grammatical markers. The word *ano* (year) is defined as a lexical marker and the words *no* and *passado* (in the last) are in the set of grammatical markers. Once the temporal patterns are established, the COP module is not used anymore.

The Annotator module is responsible for the temporal information recognition. This module uses the temporal expression patterns to identify temporal expressions and to annotate their classification in the original text. For example, in the sentence *"choveu muito no ano passado"* (It rained a lot last year), Annotator identifies the temporal expression *"no ano passado"* (last year) with a classification of *date* (see Table 3.1). This sentence is annotated as:

```
Choveu muito <EM ID="1" CATEG="TEMPO" TIPO="TEMPO_CALEND" SUBTIPO="DATA">
no ano passado</EM>.
```

The Resolver module performs the interpretation and the normalization of the temporal expressions annotated in the texts into *chronons*. This module computes the dates using a set of normalization rules. The rule to be applied is chosen based on the type of the expression (see Table 3.2). For example, the rule *doc_timestamp* -1 time_measure_unit(**y**) is used in deictic indexes, with the word *ano* (year) that defines the time measure unit. The arithmetic operations adding (+1) or subtracting (-1) is defined by the modifier marker of the expression, *próximo* (next) or *passado* (last), respectively. Considering the annotated sentence of the previous example, and the document timestamp *10-06-2015*, the Resolver output is the following sentence:

```
Choveu muito <EM ID="1" CATEG="TEMPO" TIPO="TEMPO_CALEND" SUBTIPO="DATA"
VAL_NORM="2014-XX-XX">no ano passado</EM>.
```

In this work, we use all the modules together, but each one can be used individually. Indeed, the evaluation of the effectiveness of the modules presented in Section 3.2.5 was carried out using each module independently from the others. In the development of this tool, as well as in the other tools developed, we tried to find a trade-off between efficiency and simplicity, while maintaining an appropriate level of effectiveness. In fact, the evaluation of this system has become a quite complex process due, in part, to the lack of evaluation collections in the Portuguese language for this purpose. Thus, we had to create some evaluation collections to evaluate the tool modules separately.

3.2.2 Temporal Expressions Recognition

The proposed method for recognition of temporal expressions in Portuguese texts is based on a two-stage approach, each stage being carried out by a different module: the first stage is executed by the COP, while the second one is carried out by the Annotator module (see Figure 3.4). COP creates semantically classified temporal patterns, which are later used by Annotator to identify and classify Portuguese temporal expressions, writing the annotation in the original text. These two modules are described below.

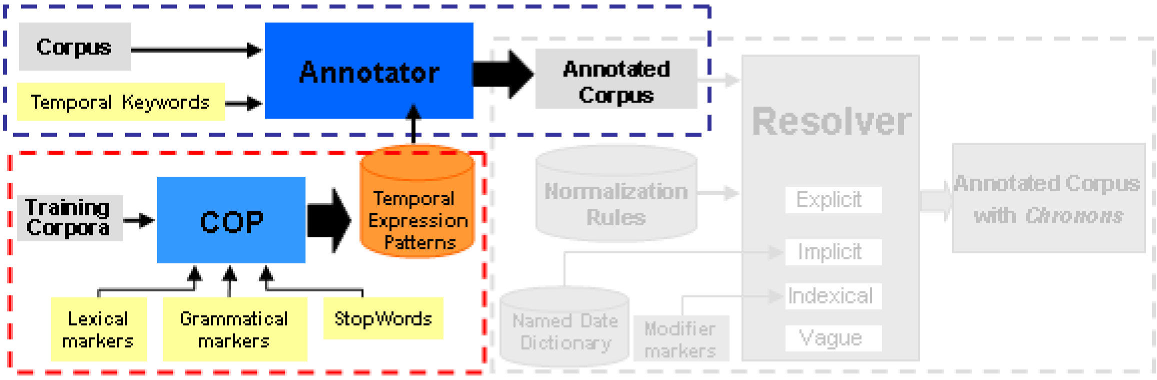


Figure 3.4: Temporal expressions recognition: Annotator and COP.

Co-Occurrence Processor Module

COP is the first module to be executed to create the patterns required by the Annotator module to identify and classify the Portuguese temporal expressions, and to perform their annotation.

Figure 3.4 shows the architecture of the COP module. This module produces semantically classified temporal patterns, based on regular expressions. The patterns are created by using word co-occurrences, determined from corpora and a pre-defined set of seed keywords derived from the temporal references of the language used. A statistical approach is used to decide which patterns must be created according to the co-occurrences and their frequency.

COP can be easily executed over various corpora, yielding a considerable number of patterns that enrich the annotation stage. It is worth noting that the COP module is only needed to get the set of temporal patterns and once the temporal patterns are established, the COP module is not used anymore. Nevertheless, the patterns can be fine tuned later on to improve the identification of temporal expressions.

The COP module has the following execution steps:

1. Determine a list of temporal expressions and their frequency
2. Prune the temporal expressions list
3. Aggregate the temporal expressions included in the list
4. Create patterns through regular expressions and associate to them the temporal expressions classification

Step#1. A list composed of the temporal expressions found in the input corpora and their frequency is created. These expressions are found by using the reference words, which are divided into two sets: lexical markers and grammatical markers. Table 3.3 presents some examples. The lexical markers are used to find their co-occurrences and must include all Portuguese words from which temporal expressions can be composed, namely months, seasons, weekdays, units of temporal measure, such as day, week, month, year, etc. A maximum distance of *n* words is established before and after the lexical marker. Following are some examples of expressions found with the lexical marker *ano* (year) and *n=2*:

(1) "No **ano** passado", "No último **ano**" (In the last **year**)

(2) "No **ano** seguinte" (In the following **year**)

(3) "No próximo **ano** de 2010" (In the next **year** 2010)

Lexical markers	months, seasons, weekdays, Natal/*Christmas*, Páscoa/*Easter*, Carnaval/*Carnival*, década/*decade*, século/*century*, ano/*year*, mês/*month*, dia/*day*, hora/*hour*, minuto/*minute*, ontem/*yesterday*, amanhã/*tomorrow*, hoje/*today*, manhã/*morning*, ...
Grammatical markers	prepositions {em/*in*, durante/*during*, desde/*since*, (n)este/*(in) this*, ...}, ordinal adjectives {anterior/*previous*, seguinte/*following*, próximo/*next*, ...}

Table 3.3: Examples of lexical and grammatical markers.

In this step, an expression is considered solely if it is composed of at least two words (*n>=1*), because an expression with just one word does not have words co-occurrence. Thus, the words alone that are a temporal expression are not included in this list. However, these words are immediately classified and included in the set of patterns, such as the temporal adverbs ending with the suffix *"–mente"*, such as *diariamente* (daily).

Step#2. The list of expressions is pruned by using the grammatical markers to remove the expressions that do not make semantic sense in a language context. A list of stopwords is also used to exclude expressions with a reference word but which are not temporal expressions in reality. For example, the reference word *Maio* (May) is in the sentence *"A rua 1.º de Maio (...)"* (The 1st May street), but the expression *"1.º de Maio"* (1st May) is not a temporal expression in this sentence because it is a street name. So, if the word *rua* (street*)* belongs to the stopwords list, this expression is excluded from the processing.

Step#3. The expressions with numbers representing a quantity, or date and time references are aggregated and marked up with a special tag, such as *tag_QUANT*, *tag_YEAR*, *tag_MONTH*, *tag_WEEKDAY*.

The aggregation is carried out according to the following rules: First, the temporal expressions are aggregated if they contain a reference. For example, the expressions *"em Janeiro"* (in January) and *"em Fevereiro"* (in February) are aggregated in a single expression with a special tag in the month position *"em **tag_MONTH**"* (in tag_MONTH).

Second and last rule, the temporal expressions are aggregated if they contain more than one co-occurrence with the same temporal word at the same position. For example, the expressions *"No **ano** passado"* (In the last year) and *"No **ano** seguinte"* (In the following year) are aggregated in an expression *"No **ano** passado | seguinte"* (In the last | following year).

The frequency of the aggregated expressions is the sum of the frequency of each expression. The resulting list is ordered by frequency (greater to less). Some expressions can be excluded by a previously defined minimum frequency threshold.

Step#4. The patterns are defined by regular expressions. For example,

```
[Nn]o ano (passado|anterior|seguinte)
```

For each pattern it is also associated the temporal classification (see Section 3.2). Following are some examples[5] of the COP output (pattern, classification):

```
(1) ([Nn]o ano (passado|anterior|seguinte), DATE)
(2) ([Dd]urante o mês de tag_MONTH, DURATION)
(3) (tag_QUANT vezes por (dia|semana|mês|ano), FREQUENCY)
```

Creating patterns is a complex task that involves an inherent subjectivity in the identification and classification of the temporal expressions. For this reason, at this stage, a manual validation is carried out to prune some incorrect patterns, and to associate the correct semantic classification. However, this is a minor problem, since a considerable number of patterns are also created and available to be used by Annotator. COP is only required to create more patterns. In any case, we intend to apply a supervised classification algorithm for future work, since now we have considerable data to create a training collection.

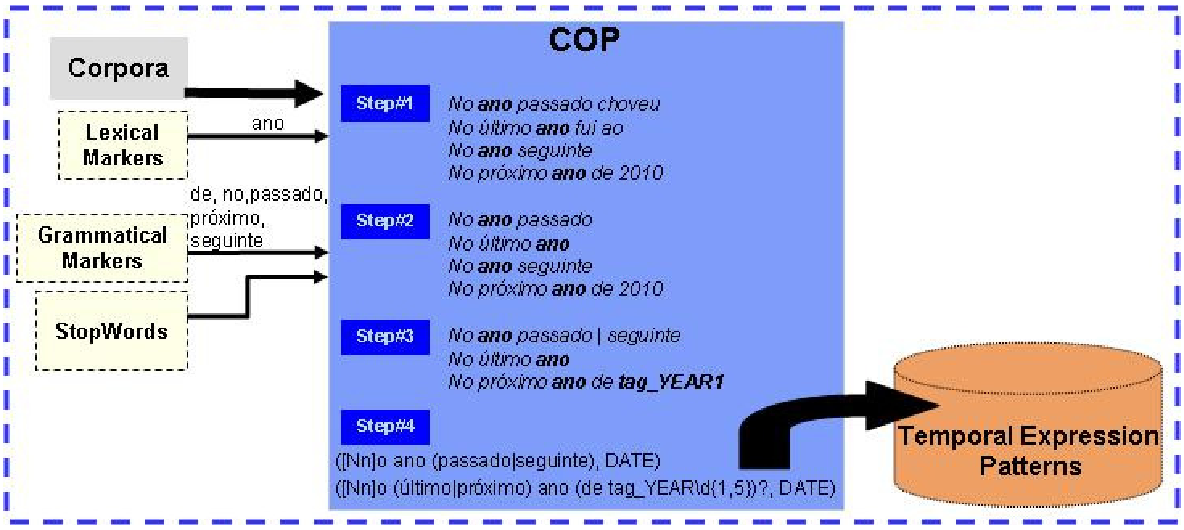


Figure 3.5: An example of patterns creation by COP module.

[5] English version**:** (1) In the (last | following) **year** (2) During the **month** of **tag_MONTH** (3) **tag_QUANT** times per (**day** | **week** | **month** | **year**)

Figure 3.5 shows an example of the step-by step patterns creation by COP Module, considering the following configuration: *n=2*, the lexical marker={*ano*} (year), and the grammatical markers={de, no, passado, próximo, seguinte[6]}.

In Step#1, COP scans the text for sentences with the lexical marker *ano* (year). It extracts expressions that contain a maximum distance of *2* (*n*) words before and after the lexical marker.

In Step#2, COP removes words that are not in the list of grammatical markers from the expressions. The word *choveu* (rained) was removed from the first expression. COP excluded *"fui ao"* (went to the) from the second expression.

In Step#3, COP aggregates expressions. The expressions *"No **ano** passado"* (In the last year) and *"No **ano** seguinte"* (In the following year) are aggregated in an expression *"No **ano** passado | seguinte"* (In the last | following year). In the last expression, *2010* are updated to *tag_YEAR1* in order to generalize the expression.

Finally, in Step#4, the regular expressions are created. For each one is also associated the semantic classification.

Annotator Module

The Annotator module processes each sentence in Portuguese texts to determine whether it matches any of the temporal patterns created by COP, and, if so, the sentence is annotated with semantic classification corresponding to the matching pattern in the original text. Besides temporal expression patterns, Annotator requires a temporal keywords list used to exclude sentences without any temporal word or date/time references from processing in order to improve the execution time. Figure 3.6 shows the components of Annotator.

The document processing carried out by the Annotator module is done sentence by sentence. So, first the text must be divided into sentences. Then, each sentence is processed in the four steps:

1. Exclude all the sentences without any temporal word or date/time references
2. Generate candidate temporal expressions
3. Match candidate expressions and temporal patterns
4. Replace the *special tags* by the original text.

[6] English version: the, in, last, next, following

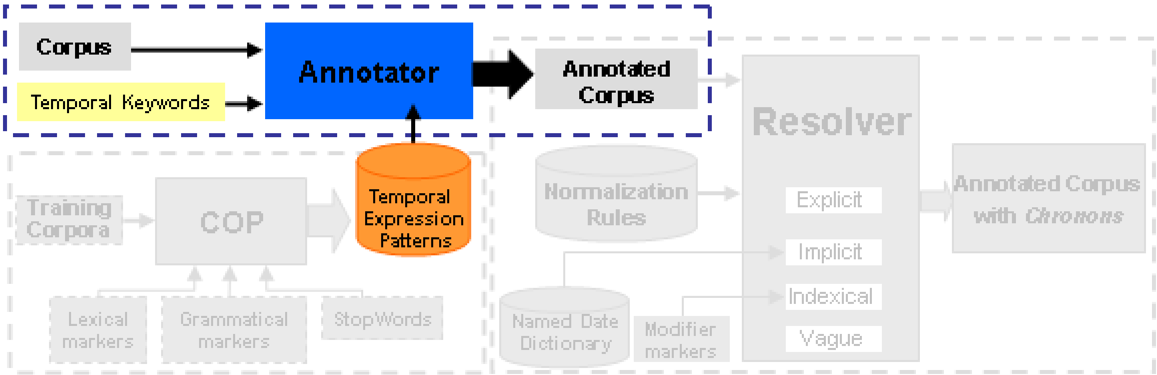


Figure 3.6: Temporal expressions recognition: Annotator.

Step#1. This step is meant to improve the performance by excluding all the sentences that cannot have a temporal expression. Only sentences with date and time references and/or temporal words from the Portuguese language are processed. The temporal words that may indicate the presence of a temporal reference must be defined in a temporal keyword list (see Figure 3.6). For example, the sentence *"Lisboa é a capital de Portugal"*[7] is not processed. However, the sentence *"**Hoje** está sol"*[8] is processed, since *hoje/today* is a temporal word.

Step#2. The candidate temporal expressions are generated. First, the temporal expressions are identified. For example, time expressions, e.g. *8:00 a.m.*, date expressions, which can be complete dates, e.g. *22-02-2009*, or incomplete dates, e.g. *Maio*[9], *Segunda-feira*, *Abril/2013*, *2008*. Then, these expressions are marked up with a "special tag", such as *tag_DATE*, *tag_MONTH*¸ *tag_YEAR*, *tag_WEEK*.

Step#3. It is verified if the candidate expressions match any temporal pattern (see Figure 3.6). In this case, each sentence is annotated in the original text with a semantic classification corresponding to the matching pattern.

Step#4. The tagged sentences are processed in order to replace the "special tags" by the original text.

The following is one example of the annotated sentence *A operação foi iniciada **ontem** às **7 horas***[10]. The Annotator module recognized two temporal expressions in this sentence, *ontem* and *às 7 horas*.

[7] English version: *Lisbon is the capital of Portugal.*
[8] English version: ***Today** is sunshine.*
[9] English version: *May*, *Monday*, *April/2013*
[10] English version: *The operation was started **yesterday** at **7 a.m.***

```
A operação foi iniciada <EM ID="1" CATEG="TEMPO" TIPO="TEMPO_CALEND"
SUBTIPO="DATA">ontem</EM> <EM ID="2" CATEG="TEMPO" TIPO="TEMPO_CALEND"
SUBTIPO="HORA">às 7 horas</EM>.
```

Figure 3.7 shows the step-by-step annotation of the Portuguese sentence *A missão científica da nave foi concluída em 30 de Abril de 2002*[11] by the Annotator module, according to the description presented above. Appendix A shows a full example of the text processing by Annotator.

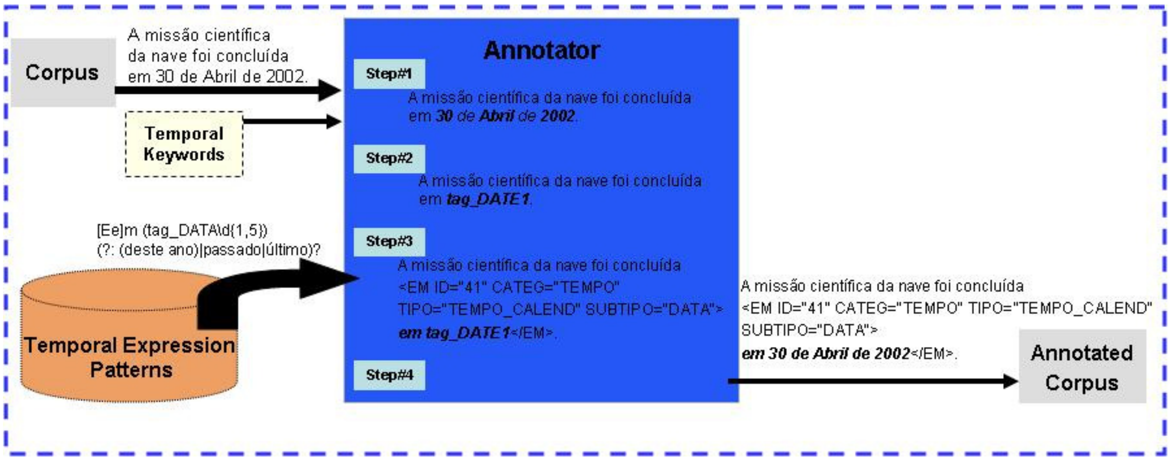

Corpus
A missão científica
da nave foi concluída
em 30 de Abril de 2002.
Temporal
Keywords
[Ee]m (tag_DATA\d{1,5})
(?: (deste ano)|passado|último)?
Temporal Expression
Patterns
Annotator
Step#1
A missão científica da nave foi concluída
em 30 de Abril de 2002.
Step#2
A missão científica da nave foi concluída
em tag_DATE1.
Step#3
A missão científica da nave foi concluída
<EM ID="41" CATEG="TEMPO"
TIPO="TEMPO_CALEND" SUBTIPO="DATA">
em tag_DATE1</EM>.
Step#4
A missão científica da nave foi concluída
<EM ID="41" CATEG="TEMPO" TIPO="TEMPO_CALEND"
SUBTIPO="DATA">
em 30 de Abril de 2002</EM>.
Annotated
Corpus

Figure 3.7: Example of a Portuguese sentence annotated by the Annotator module.

In Step#1, the sentence is not excluded from the processing, since it includes a date.

In Step#2, since the sentence contains a date, the sentence is marked up with the special tag *tag_DATE*, replacing the original text date *30 de Abril de 2002*.

In Step#3, the sentence matches the temporal pattern *em tag_date* (in tag_date) with the classification DATE. The sentence is annotated as:

```
A missão científica da nave foi concluída <EM ID="1" CATEG="TEMPO"
TIPO="TEMPO_CALEND" SUBTIPO="DATA">em tag_DATE</EM>.
```

In Step#4, the original text of the date included in the sentence is replaced. The result is:

```
A missão científica da nave foi concluída <EM ID="1" CATEG="TEMPO"
TIPO="TEMPO_CALEND" SUBTIPO="DATA">em 30 de Abril de 2002</EM>.
```

[11] English version: *The scientific mission of the spacecraft was finished on April 30, 2002.*

3.2.3 Temporal Expressions Resolution

The resolution of temporal expressions is performed by the Resolver module. The objective is to map the temporal expressions found in the document content into a normalized format, as far as possible, since it is not possible to normalize all the temporal expressions.

The normalized format is a discrete representation of time, denoted *chronon* by Alonso et al. (2009b). A *chronon* is a normalized date which is anchored in a calendar or clock system, considering the four timelines *T* = *{Th, Td, Tm, Ty}* for hours, days, months, and years, respectively. The base timeline, *Td,* is a timeline with day granularity, defining an interval of sequential day *chronons*. In this manner, intervals are normalized with two *chronons*, one for each interval boundary.

Chronons are represented in a standard format *YYYY-MM-DDThh:mm:ssZ*, specified by ISO 8601:2004. For example, *"Maio 1, 2011"* (May 1, 2011) is normalized as *"2011-05-01"* and can be anchored in *Td*. The algorithm uses the *X* placeholder for incomplete expressions. For example, *"Junho de 2000"* (June of 2000) is normalized as *"2000-06-XX"* and since it cannot be anchored in the base timeline, a more coarse-grained timeline is used, such as *Tm* or *Ty*.

The architecture of the Resolver module is displayed in Figure 3.8. This module relies on a set of rules, used to interpret temporal expressions previously annotated by the Annotator module. Expressions classified as duration or frequency, are not yet normalized, in this version of the tool.

The rules are defined according to the way as the temporal expression is expressed in the text (explicit, implicit, deictic or anaphoric). For example,

```
(1) "ontem" (yesterday)          doc_timestamp -1 time_measure_unit(d)
```

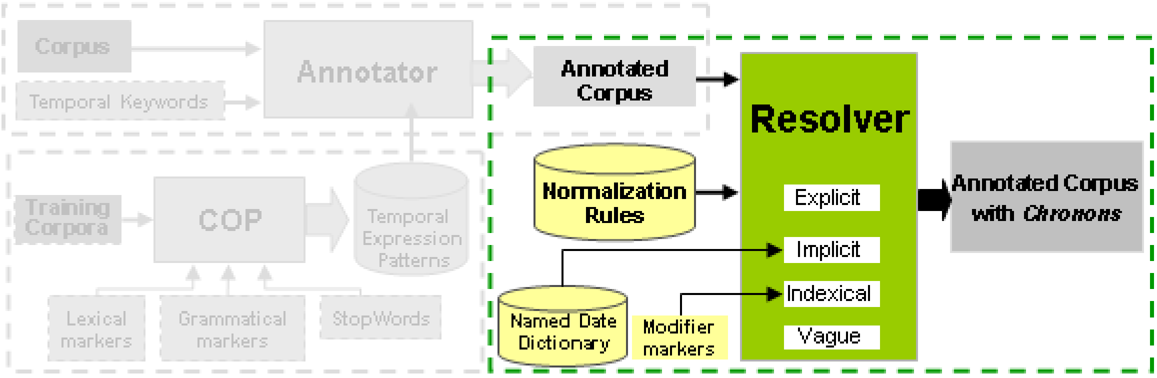


Figure 3.8: Temporal expressions recognition: Resolver.

The resolution rules defined for the indexical references must include the reference date, which can be the document timestamp or a date found in the text, and the quantity of units to add or subtract to the reference date. In the example presented above, the reference date is the document timestamp and the operation is the subtraction of a unit (-1), which in this case is the day (d).

The Resolver module starts with the document timestamp normalization. This date, a time related metadata of a document, such as the creation or publication date of the document, is very important to resolve some type of temporal references, such as *deictic timexes*.

Explicit expressions have a reduced processing, since they do not need to be submitted to the interpretation task. The processing of such expressions is only to put them into the chosen format standard. The other expressions also need to be processed to make them explicit expressions, anchoring them in a timeline, whenever possible.

The ability to resolve the implicit references and anchor them in a timeline relies on underlying time ontology of the approach used. This time ontology must have a list of the implicit temporal expressions and their corresponding explicit temporal expression. In our system, we defined a named date dictionary where this information is stored. For example, *"Dia de Natal, 2010"* (Christmas Day*)* can be represented as an explicit reference, such as *"25 de Dezembro de 2010"* (25th December of 2010), which is normalized as *"2010-12-25"*.

The process of the indexical references resolution is more complex for two main reasons. First, these references need one of two different reference dates, the document timestamp (*deitic timexes*) or the other time reference evoked in the text (*anaphoric timexes*) to be resolved. Second, these references can mention a past, present or future event. The correct rule to apply for each of these cases is chosen by modifiers of the indexical references, such as *próximo* (next), *anterior* (previous), *depois* (after), *antes* (before) or *seguinte* (following). For this phase of the work, the *anaphoric timexes* are correctly resolved only if they contain a modifier marker.

Table 3.4 shows some examples of the resolution rules used by the Resolver module in the resolution of indexical references. In the rules, *doc_timestamp* corresponds to the document timestamp, which is used in *deitic timexes*. The *anaphoric timexes* rules use *text_date* to represent the time reference evoked in the text.

Temporal Expression		**Resolver Rules**
Portuguese	English	
próximo ano	*next year*	*doc_timestamp* +1 time_measure_unit(**y**)
mês anterior	*previous month*	*doc_timestamp* –1 time_measure_unit(**m**)
há QUANT meses	*QUANT months ago*	*doc_timestamp* –QUANT time_measure_unit(**m**)
QUANT meses depois	*QUANT months after*	*text_date* +QUANT time_measure_unit(**m**)
próximo DIASEMANA	*next WEEKDAY*	*doc_timestamp* +1 time_measure_unit(**w**)
DIASEMANA passado	*last WEEKDAY*	*doc_timestamp* –1 time_measure_unit(**w**)
ontem	*yesterday*	*doc_timestamp* –1 time_measure_unit(**d**)
anteontem	*the day before yesterday*	*doc_timestamp* –2 time_measure_unit(**d**)
amanhã	*tomorrow*	*doc_timestamp* +1 time_measure_unit(**d**)
QUANT dias antes	*QUANT days before*	*text_date* –QUANT time_measure_unit(**d**)

Table 3.4: Resolution rules of the Resolver module: some examples.

Figure 3.9 shows a temporal expression resolution of the sentence *Ontem esteve um dia lindo*[12] by Resolver, considering the description presented above. This module starts with the rule identification to resolve the temporal expression *ontem* based on its type, which is deictic without any modifier marker. Then, it computes the *chronon* associated to the temporal expression, using the identified rule, and normalizes it. Finally, the annotation is updated with the normalized value *VAL_NORM=”2001-07-06”*, considering *2001-07-07* as the document timestamp.

Appendix A shows a full example of the text processing by the Resolver module.

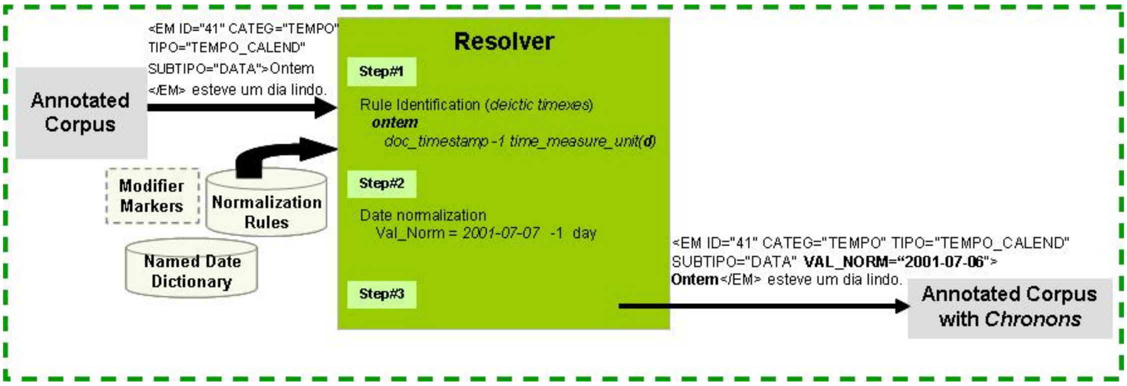

Annotated Corpus
<EM ID="41" CATEG="TEMPO" TIPO="TEMPO_CALEND" SUBTIPO="DATA">Ontem </EM> esteve um dia lindo.
Modifier Markers
Normalization Rules
Named Date Dictionary
Resolver
Step#1
Rule Identification (deictic timexes)
ontem
doc_timestamp -1 time_measure_unit(d)
Step#2
Date normalization
Val_Norm = 2001-07-07 -1 day
Step#3
<EM ID="41" CATEG="TEMPO" TIPO="TEMPO_CALEND" SUBTIPO="DATA" VAL_NORM="2001-07-06"> Ontem</EM> esteve um dia lindo.
Annotated Corpus with Chronons

Figure 3.9: Sample of a date resolution by the Resolver module.

[12] English version: *yesterday it was a beautiful day*.

3.2.4 Preliminary Evaluation of Annotator at the Second HAREM

PorTexTO is the Annotator module developed in Perl language following the algorithm presented in Section 3.2.2. This module previously divides text into sentences by using the Perl module Lingua::PT::PLNbase[13], in order to do a sentence by sentence processing [Craveiro et al., 2008].

A first version of this tool, PorTexTO 1.0 participated in the second edition of HAREM, named as Second HAREM. HAREM[14] is the first evaluation contest, organized by Linguateca [lin, nd] for Named Entity Recognition (NER) for Portuguese [Mota and Santos, 2008].

The Second HAREM was organized in several categories, such as LOCAL (place), ACONTECIMENTO (event), PESSOA (person), TEMPO (time), etc. However, only TEMPO is the most suitable category to the goals of this work. PorTexTO 1.0 participated solely in TEMPO category but in the two tasks: identifying and semantic classification of entities mentioned.

Considering the requirements of the Second HAREM, PorTexTO 1.0 presents some limitations. PorTexTO 1.0 does the classification using only TIPO (type), and SUBTIPO (subtype) of the TEMPO category. The subtype INTERVALO (interval) of the type TEMPO_CALEND and the GENERICO (generic) type were excluded from the classification task.

Note that in the Second HAREM only the Annotator module of the extraction was evaluated. This means that in this version PorTextO only identifies and classifies temporal expressions. The normalization of the expressions is not developed in this version of the PorTexTO. For this reason, in this participation, it was not employed neither the time extended classification nor the normalization tasks.

The temporal expressions patterns used by PorTexTO 1.0 only matched simple temporal expressions, which are composed of a single temporal reference.

The complex temporal expressions, such as *"no **dia** 10 do **mês** passado"* (on the 10th of last month) were treated by PorTexTO 1.0 as two simple expressions; so, the semantic classification is not considered accordingly correct as the temporal guidelines defined by Second HAREM organization [Hagège et al., 2008a].

The temporal expressions patterns were created by the COP module by using the lexical and the grammatical markers previously presented in Table 3.3. The COP was configured with *n=2*,

[13] Available at http://search.cpan.org/~ambs/Lingua-PT-PLNbase-0.20 [March 15th, 2008]

[14] ***H**AREM é uma **A**valiação de **R**econhecedores de **E**ntidades **M**encionadas*

meaning that the maximum length of a pattern is five words. This module used in the input the following corpora, available at Linguateca site [lin, nd]:

- the HAREM Collection version 2.0, used in the two events of the First HAREM evaluation contest. This collection is composed of 1202 documents, with approximately 520,000 words. The documents are in different textual genres, such as technical, political, literary, expository, journalistic interviews, electronic mails and Web pages. The details of this collection were published by Santos and Cardoso (2007).

- the Second HAREM Collection (henceforth HC), with documents in different textual genres, such as journalistic, educational, blog, question, FAQ, dialog, literary, juridical and advertising. It is composed of 1040 documents with 668,817 words (see Table 3.5). This collection was detailed by Mota et al. (2008a).

The results presented in this section are only focused on the scenario used by the PorTexTO 1.0, scenario with TEMPO (time) category, named TEMPO (time) scenario. The evaluation in other scenarios also included other categories that were not recognized by the tool. The results discussion is based on two collection used by the TEMPO scenario evaluation: Gold Collection of Second HAREM (henceforth GC) and Gold Collection of TEMPO category (henceforth TGC). Note that the PorTexTO 1.0 evaluation uses the TGC, but it does not take into account the specific attributes of the TEMPO category used for the time extended classification and the normalization, which were not carried out by the tool, as previously explained.

	# Collection documents	# Collection sentences	# Collection words
Second HAREM Collection (HC)	1040	33,712	668,817
Gold Collection (GC)	129	4,465	74,350
Time Gold Collection (TGC)	30	622	12,992

Table 3.5: Second HAREM collections statistics.

Table 3.5 presents statistical information about all the collections used by the evaluation contest of HAREM. These collections were described by Mota et al. (2008a) and are available at Linguateca site [lin, nd]. A temporal characterization and a complete description of the Second HAREM Collection (HC) are presented in Section 4.2.

The Second HAREM organization was responsible for carrying out the evaluation of the participating systems. The details of this evaluation, as well as the results obtained by all systems were published by Mota et al. (2008b). The results of the identification task were calculated by using the traditional measures, precision, recall, and harmonic mean F, also named F-measure, whose formulae are presented below:

$$\Pr ecision = \frac{TP}{TP + FP} \tag{3.1}$$

$$\operatorname{Re} call = \frac{TP}{TP + FN} \tag{3.2}$$

$$F - measure = 2 * \frac{\Pr ecision * \operatorname{Re} call}{\Pr ecision + \operatorname{Re} call} \tag{3.3}$$

In Formula 3.1 and Formula 3.2, True Positives (TP) is the number of expressions correctly identified as temporal expressions, False Positives (FP) is the number of expressions incorrectly identified as temporal expressions and False Negatives (FN) is the number of temporal expressions that were not identified by the system.

The classification task is more complex, since it has more attributes of the annotation scheme to validate, namely the correct values in category, type and sub-type. Thus, a single measure, called semantic classification combined, was defined to the correct temporal expression identified [Mota et al., 2008b]. This measure gives a score that represents the combination of the temporal expression identification and the values put in those 3 attributes (i.e., category, type and sub-type). This score is used in computation of precision and recall, as follow:

$$\Pr ecision = \frac{\sum_i score(te_i)}{Max(score(te))} \tag{3.4}$$

$$\operatorname{Re} call = \frac{\sum_i score(te_i)}{Max(scoreGC)} \tag{3.5}$$

In Formula 3.4 and Formula 3.5, *score(te)* is the score of each temporal expression identified by the system and *scoreGC* is the maximum score of the gold collection.

PorTexTO 1.0 participated in the Second HAREM with four runs. The first run, PorTexTO_1, was only useful to validate the submission process and to verify some possible incongruities in the named entities annotation. Indeed, there was not any incongruity in the annotation. Due to the objectives of this run, the results were not considered important.

The other runs, PorTexTO_2, PorTexTO_3 and PorTexTO_4, had minor differences in the definition of regular expressions of the patterns, namely (1) more coverage and less accurate; (2) less coverage and more accurate, such as:

```
(1)  [Nn]o \w+ ano                          (In the \w+ year)
(2)  [Nn]o (passado|último) ano        (In the (past|last) year)
```

Thus, the objective of these runs was to analyze the precision and recall variation when the coverage of the regular expressions was increased. Precision and recall are measures of effectiveness, in other words, they measure the quality of the system answer. In this case, the precision gives the proportion of correct annotation of the temporal expressions in all temporal expressions annotated by the system. The recall gives the percentage of correct temporal expressions annotated by the system.

Figure 3.10 and Figure 3.11 illustrate the evaluation of PorTexTO 1.0 in the four runs submitted using the two gold collections (GC and TGC). The four runs had very similar results with minor differences. However, the PorTexTO_1 run had the greater difference, since this run was submitted before the complete configuration of the tool, as previously explained. In the results obtained by the PorTexTO_2, PorTexTO_3 and PorTexTO_4 runs, the precision value was not penalized by the patterns definition with more coverage.

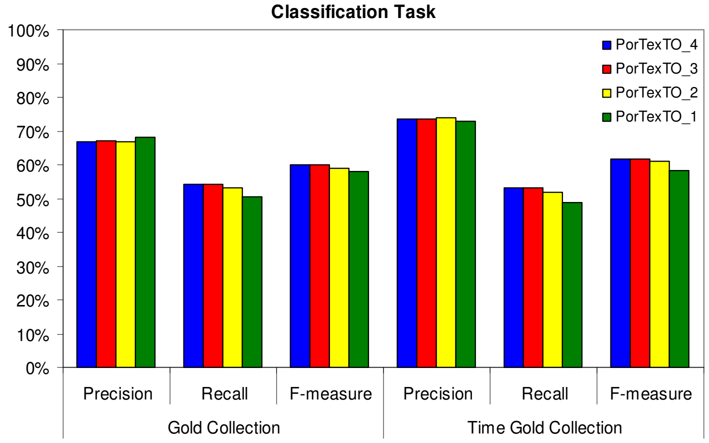


Figure 3.10: PorTexTO classification task results for GC and TGC.

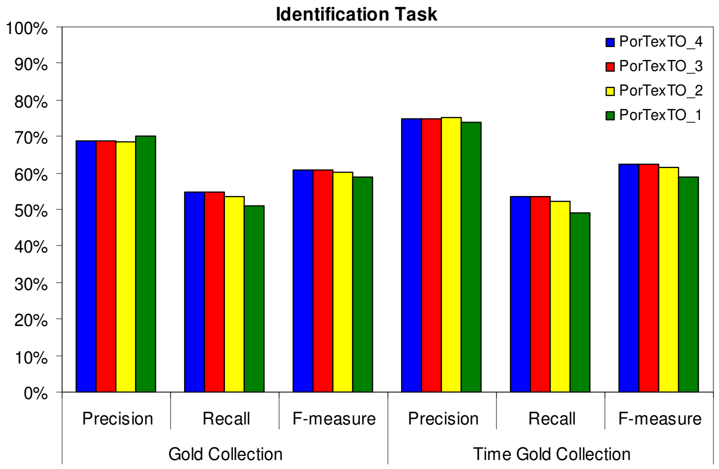


Figure 3.11: PorTexTO identification task results for GC and TGC.

	Classification Task			Identification Task		
	Precision	Recall	F-measure	Precision	Recall	F-measure
GC	0.6694	**0.5419**	0.5990	0.6871	**0.5470**	0.6091
TGC	**0.7350**	0.5327	**0.6177**	**0.7470**	0.5345	**0.6231**

Table 3.6: PorTexTO_4 results in TEMPO category.

The results obtained in the classification and identification tasks are also very similar, meaning that the temporal entities identified by PorTexTO 1.0 were well classified. As well as, the results obtained in the GC and TGC collections are also very similar. The PorTexTO 1.0 achieved a very small improvement in the precision value when it was evaluated with the TGC, but that is not significant; the difference was only approximately 6%. Table 3.6 only presents the results obtained by PorTexTO_4, since the three runs PorTexTO_2, PorTexTO_3 e PorTexTO_4 had identical results.

The PorTexTO_4 run got the 5th position in the evaluation with TGC, but the first four positions belong to the same system. However, the significant difference between the XIP-L2F/Xerox_3 run of the system on the first position was only obtained in the recall value. This means that this system recognizes more temporal expressions than PorTexTO 1.0. As we previously mentioned, PorTexTO 1.0 did not recognize some temporal expressions that are in the time guidelines of the Second HAREM. The best system was XIP-L2F/Xerox with a precision of approximately 75% for a recall of 78%. Figure 3.12 compares the results obtained by the PorTexTO_4 and XIP-L2F/Xerox_3 runs.

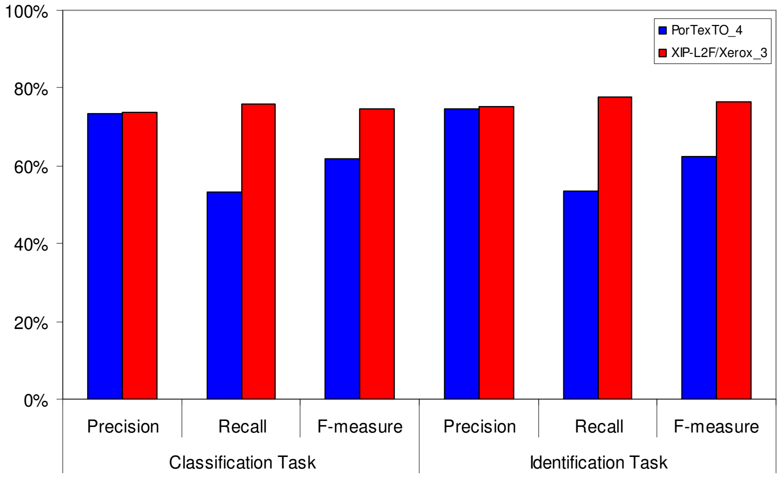


Figure 3.12: TEMPO category in TGC: PorTexTO_4 and XIP-L2F/Xerox_3 results.

In the first participation in the evaluation contest, PorTexTO 1.0 obtained auspicious results, which exceeded our expectations because this version had several limitations as previously mentioned. This tool was specially created for the participation in the Second HAREM. The short time available was insufficient for a proper implementation and test of the different modules.

Although the efficiency was not evaluated by the Second HAREM, we computed some measures. In terms of computational performance, PorTexTO 1.0 annotated HC, which was the test collection, at a rate of approximately 22KB per second. The processing of the approximately 675,000 words took three minutes and twenty seconds. The execution time achieved is quite acceptable for the objectives defined, such as the incorporation in an information retrieval system, such as *ad hoc* retrieval. Note that the processing was in a Personal Computer without any hardware improvement. The computer has 1GB RAM memory and an Intel Core 2 E6600 2.4GHz processor, running with Microsoft Windows XP Professional version 2002 SP 2.

3.2.5 Evaluation of the Extraction Approach

Knowing that there is a huge amount of documents to process in common application scenarios, one of the key decisions is to achieve the best trade-off between efficiency and effectiveness in temporal expressions extraction. As our option is to favor efficiency to some extent, with the used configuration, the system may not find all temporal expressions. In fact, the identification of temporal expressions is not an easy task and even for a trained human reader it would be difficult to identify all temporal expressions, as the notion of time is often subtly embedded in the text.

All experiments were performed in a Personal Computer with 1GB RAM memory and an Intel Core 2 E6600 2.4GHz processor, running with Microsoft Windows XP Professional version 2002 SP 2.

This section presents the evaluation sessions performed using the proposed approaches for recognition and resolution of the temporal expressions, namely the evaluation of the *Annotator* and the *Resolver* modules. The evaluation performed focused on the effectiveness of the proposed approaches. However, the efficiency was also considered in the evaluation of PorTexTO 2.0, the Annotator module, by computing of the execution time. Note that the evaluation of each module was carried out in an independent manner. Each module was separately evaluated from the other one. The results obtained are also published [Craveiro et al., 2009, Craveiro et al., 2010, Craveiro et al., 2012].

The collections used are based on the Second HAREM Collection (HC), presented in Section 3.2.4. The following is the description of the setting used in the experiments and the results obtained.

Collections

There is not much choice, due to the lack of resources for Portuguese language processing. Thus, to the purpose of the evaluations tasks, the HC collection created by Linguateca for the Second HAREM Evaluation Contest, and available at Linguateca site [lin, nd], is the only available collection. HC is composed of documents in the Portuguese language, so it is the corpus chosen to carry out all the experiments of this work. The texts of the HC collection are structured in different literary genres and are written in two variants of the language: European Portuguese and Brazilian Portuguese. A detailed description of this collection is presented in Section 4.2.

The HC collection includes two subsets both used as the evaluation collections, GC and TGC.

The TGC subset was used to evaluate the Annotator module. TGC is composed of 30 documents, 622 sentences and 12,992 words. These documents were annotated manually, taking into account the time HAREM guidelines [Hagège et al., 2008a] (see Section 3.2). The test set, named ATC, was another subset of the HC, which is composed of all documents of HC that do not belong to the TGC. ATC is also used in the Annotator evaluation. The COP module also used the ATC subset to create the temporal expressions patterns to be used in the Annotator evaluation.

The GC subset was used in the Resolver evaluation. This subset consists of 129 documents with 4,465 sentences and 74,350 words, which have also been annotated manually, following time HAREM guidelines [Hagège et al., 2008a] (see Section 3.2). Since there was no collection to evaluate the task of temporal expressions resolution, we had to create one collection, named

Resolver Evaluation Collection (REC). Therefore, this evaluation set is composed of 30 documents with 7,125 words whose documents were randomly extracted from HC and were also manually annotated and normalized.

To summarize, Table 3.7 shows information about all collections used in the evaluation.

Module		Collection	# Collection documents	# Collection words
Annotator		Second HAREM Collection (HC)	1040	668,817
	Evaluation Set	Time Gold Collection (TGC)	30	12,992
	Test Set	Annotator Test Collection (ATC)	1010	655,825
Resolver	Evaluation Set	Resolver Evaluation Collection (REC)	30	7,125
	Test Set	Gold Collection (GC)	129	74,350

Table 3.7: Statistical information for evaluation tasks.

Annotator Evaluation

The primary goal of the Annotator Evaluation was to evaluate the performance of the method in a restricted environment. Therefore, we configure the COP only to create patterns of simple temporal expressions, expressions composed of only one temporal reference, such as a temporal word, a date or a time, and a maximum of *n=2* words before and/or *n* words after. This means that the length of a temporal expression is between 1 and 5 words.

The COP was configured with *n=2* and limited to the following lexical markers: *months*, *seasons*, *weekdays*, most important holidays, such as *Natal* (Christmas), *Páscoa* (Easter) and *Carnaval* (Carnival) and the following words[15]: *década*, *século*, *ano*, *mês*, *semana*, *dia*, *hora*, *minuto*, *ontem*, *anteontem*, *amanhã*, *hoje*, *manhã*, *noite*, *tarde*. Furthermore, a set of limited grammatical markers[16] were included in the temporal patterns. This set is composed of prepositions, and their contraction with pronouns *{à(s), de, em, durante, desde, pelas, no, naquele, neste, nesse, este, esse}*, ordinal adjectives *{anterior, seguinte, próximo, passado, último}* and *haver* (to have) verb conjugations. Note that in the pruning step, the stopwords are not considered yet.

[15] English version: *decade, century, year, month, week, day, hour, minute, yesterday, the day before, tomorrow, today, morning, night, afternoon.*

[16] English version: prepositions *{in, the, during, for, since, by, in this, in that, this}*, ordinal adjectives *{previous, following, next, past, last}*

The set of experiments were carried out by PorTexTO 2.0, the Annotator module developed following the approach described in Section 3.2. Some improvements were performed in PorTexTO 1.0, the tool version that participated in the evaluation contest Second HAREM (see 3.2.4).

The experiments carried out to the evaluation of the Annotation approach were divided in two tasks: identification and classification. The goal of the Identification task was to recognize the complete temporal expressions, while the Classification task was focused on the correct specification of the type and subtype parameters of the expressions classification.

In order to clarify these tasks, some examples are presented below. In Example (1), the expression *"1909-1955"* is correctly identified but incorrectly classified. The subtype must be INTERVALO (interval) instead of DATA. In Example (2), the classification is correct but the expression *"2009"* is incomplete. Following the time guidelines [Hagège et al., 2008a] (see Section 3.2), the correct identification must be *"**em** 2009"* (in 2009).

```
(1) (...)CATEG="TEMPO" TIPO="TEMPO_CALEND"  SUBTIPO="DATA">1909-1955</EM>
(2) em  <(...)CATEG="TEMPO" TIPO="TEMPO_CALEND"  SUBTIPO="DATA">2009</EM>
```

The three usual metrics were defined for the effectiveness evaluation of Annotator, using the TGC evaluation set: precision, recall, and harmonic mean F (F-measure). The formula used to calculate the classification was defined by Oliveira et al. (2008). The measure used for efficiency purposes was the time spent on the Annotator module to identify and classify the temporal expressions in the HC test set, named *execution time*.

Results of the Annotator Evaluation

The COP created 289 patterns by processing the ATC test set. Indeed, these 289 patterns can detect more than 289 different expressions, because some of them have more than one combination of temporal expressions. Note also that approximately 17% of these patterns are entirely used in the identification of different formats of dates and times.

The execution time was calculated in two scenarios: Scenario 1 – the first step of the Annotator module was skipped, but all the sentences are processed by every other steps of this module; Scenario 2 – all steps are executed, therefore, only the sentences which we believe could indicate the presence of a temporal reference are processed. The full details of the Annotator steps are in Section 3.2.2.

Table 3.8 shows the results obtained in both scenarios. In scenario 2, only 17,525 of 33,712 sentences, about 52%, proceed to the next step; the processing finishes here to the other 16,187 sentences. The execution time decreases about 27.5% justified by the missing pattern matching step with the remaining sentences. This way, the performance was improved by using the first step of the Annotator module. This module processed the test collection with an output rate of about 22KB per second.

Annotator Module	# Collection sentences	# Sentences processed	Execution Time (seconds)
Scenario 1	33,712	33,712	262
Scenario 2	33,712	**17,525**	**190**

Table 3.8: Annotator module: results in the two scenarios.

	Identification Task		Classification Task	
	PorTexTO 2.0	XIP-L2F/ Xerox system	PorTexTO 2.0	XIP-L2F/ Xerox system
Precision	84,27%	75,31%	83,05%	73,76%
Recall	64,10%	77,59%	64,23%	75,80%
F-measure	72,82%	76,43%	72,44%	74,77%

Table 3.9: Annotation results: PorTexTO 2.0 *versus* XIP-L2F/Xerox.

The results of the effectiveness are computed using the tool SAHARA[17], available at Linguateca site [lin, nd].

Annotator obtained promising results in the Classification and Identification tasks by using the HC test collection and the TGC evaluation collection. Table 3.9 shows the effectiveness results obtained in the evaluation collection. The results of the two tasks obtained by the Annotator module do not have significant differences, which mean that this module shows the same behavior in the two tasks. In conclusion, if this module identifies a given temporal expression, then it will achieve a good success in its subsequent classification.

Table 3.9 also shows the results obtained by the XIP-L2F/Xerox system [Hagège et al., 2008b] using the same collections. This system was ranked in first place in the Second HAREM [Mota et al., 2008b, Hagège et al., 2008b]. Our approach matches the results of the top system concerning

[17] http://www.linguateca.pt/harem/ [September 20th, 2015]

precision, but it shows lower recall. This is mainly due to the restricted set of lexical and grammatical markers used by COP to generate the patterns, which affects the recall. However, we believe that the recall can be improved by increasing the restricted set used by the COP module. We plan to exploit this in future work.

Although COP was configured with *n=2*, which means that the expressions were limited to 5 words, only approximately 12% of the TGC expressions have more than two words before and/or after the lexical marker. Table 3.10 shows the number of temporal expressions found in the TGC evaluation collection, considering the variation of *n* between 2 and 7. There are very few temporal expressions – only about 1%, when *n>4*.

	# words before and/or after the lexical marker					
	n=2	n=3	n=4	n=5	n=6	n=7
# temporal expressions	205	18	8	1	1	0

Table 3.10: Temporal expressions in TGC, with $2 \leq n \leq 7$.

In the precision calculation of the Identification task, the temporal expressions partially correct are not considered. PorTexTO 2.0 found 178 temporal expressions, 150 of which are correctly identified. Indeed, the incorrect expressions are only 3 of 28; the others are incomplete because one or more words are missing in the annotated expression.

Figure 3.13 shows the precision and the recall values when the variation of *n* is between 2 and 7. These values were calculated by analyzing the missing temporal expressions or the temporal expressions incorrectly identified by PorTexTO 2.0. This analyze was manually performed, taking into consideration the number of words of the missing and incorrect expressions annotated by PorTexTO 2.0.

We verified that the precision improves significantly when *n* goes from 2 to 3, as well as the recall value. Improvement is still seen from *n>3*, but at a lower rate. The best values of precision and recall are 96.35% and 79.06%, respectively. This means that if the COP creates temporal patterns with *n>2* and one more temporal word, i.e., lexical marker, the values of precision and recall are improved. The recall achieved is not as good as the precision, but nevertheless the recall reached about 80%. As mentioned before, the improvement of this metric can be done by increasing the restricted sets of lexical and grammatical markers used by the COP module.

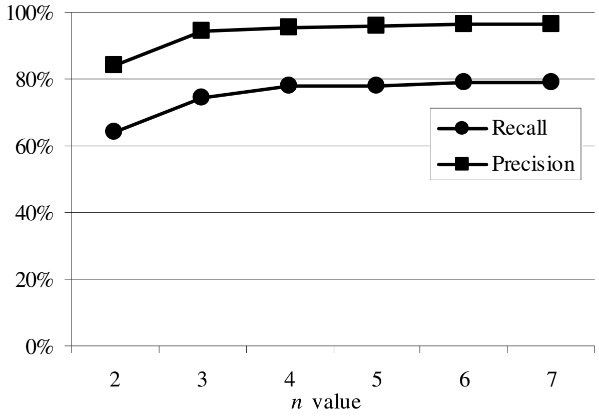


Figure 3.13: Precision and recall values with *2≤n≤7*.

Resolver Evaluation

The effectiveness of the Resolver approach, presented in Section 3.2.3, was the focus of the evaluation performed with the GC test set, and the REC evaluation set (see Table 3.7).

In this evaluation, a correctly normalized date must have all the fields (Year, Month and Date) well filled in the *chronons* representation; otherwise, it is considered incorrect. Table 3.11 shows some examples of the *chronons* evaluation, where CORR and INCO, represent correct and incorrect resolutions, respectively. A temporal expression that can be mapped into a *chronon*, but was not resolved by Resolver is classified as MISS.

Manual Normalization	Resolver module	Classification
VAL_NORM="1993-12-31"	VAL_NORM="1993-12-31"	CORR
VAL_NORM="1994-01-01"	VAL_NORM="1994-01-XX"	INCO
VAL_NORM="1995-XX-XX"	VAL_NORM=""	MISS

Table 3.11: Resolver module: some examples of the *chronons* evaluation.

Table 3.12 reports the results obtained by our module, which are also published [Craveiro et al., 2010]. The columns CORR and INCO respectively report the number of *chronons* correctly and incorrectly resolved. The number of *chronons* that was not resolved by the Resolver module, but that occurs in the corpus is reported as MISS. In this evaluation, there was not spurious *chronons* since the input data of this module is a corpus with temporal expressions already identified and classified. Accuracy is given by Formula 3.6.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.6}$$

In Formula 3.6, True Positives (TP) is the number of temporal expressions correctly resolved (CORR), False Positives (FP) is the number of temporal expressions incorrectly resolved (INCO) and False Negatives (FN) is the number of temporal expressions that were not resolved by the system (MISS). As we mentioned above, the number of spurious *chronons* (TN) is 0.

CORR	**MISS**	**INCO**	**Accuracy**
273	10	78	75.62%

Table 3.12: Resolver module: evaluation results.

The results obtained revealed a good performance of this module, even with some limitations, namely in the indexical references. Obviously, there is still room for improvements. The temporal expressions classified as duration or frequency, are not yet normalized. A source of errors is due to the incorrect rules that system applied to some of the indexical references. In this stage of the work, indexical references are well resolved if they contain a modifier marker. Moreover, the module did not yet complete the dates with the pattern *XXXX-mm-dd*, for instance *"XXXX-06-01"*. This situation occurred in 40 *chronons*, representing about 10% of total. Year is the missing field of these incomplete dates. These dates can be completed in two different ways, according to the type of the temporal reference associated to them (see Table 3.2): extracting the missing fields from the document timestamp, if the expression is a *deictic timex*; or obtaining the missing fields from the other temporal references near the incomplete date, if the expression is an *anaphoric index*.

3.3 Temporal Segmentation of Texts

The overall objective of this work is to enrich the information retrieval indexes with temporal information by using segmentation of texts based on temporal discontinuities. The approach presented in the previous sections provides the normalized temporal information used in the definition of the temporal text segmentation approach presented in this section.

In text segmentation, the text is divided into smaller units taking into account the boundaries of topic changes. Text segments are a set of contiguous sentences or paragraphs sharing the same topic [Hearst, 1994, Beeferman et al., 1999, Misra et al., 2009].

Our temporal text segmentation approach is based on text segmentation by topic. The boundaries of a segment are firstly defined by the temporal information of the segment, unless this information does not exist in the text. In that case, the boundaries are identified by the topic changes. A temporal segment is defined as a set of adjacent sentences that shares the same temporal focus. The temporal segments are also tagged with the timestamps to later provide an association of words and dates. Indeed, the timestamps are the *chronons* collected from the text that belongs to the segment.

Figure 3.14 shows the architecture of the tool developed, named *Time4Word*, to allow the temporal segmentation of Portuguese texts. The tool uses the temporal information of documents, metadata and contents, to partition text into temporally coherent segments. The *chronons* found in the texts give the boundaries of the segments. However, there are many sentences without *chronons*. In this case, it is verified if there is any continuity or discontinuity marker to keep the sentence in the segment or to start a new segment, respectively. It is computed the similarity of sentences without *chronons* and markers, which is based on the topic segmentation. A threshold is defined to determine if the sentences are similar or not. The stopwords list is used to simplify the sentences insofar as these words do not add information to the topic definition. The timestamp of the segments are annotated in the input text.

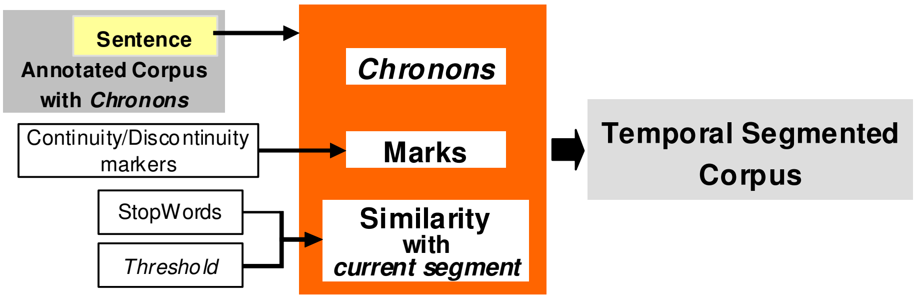

Sentence
Annotated Corpus with *Chronons*
Continuity/Discontinuity markers
StopWords
Threshold
Chronons
Marks
Similarity with *current segment*
Temporal Segmented Corpus

Figure 3.14: System architecture for temporal segmentation.

The important input data for this temporal segmentation module are the collected *chronons* by Resolver, a module of the *Extraction* tool (see Section 3.2.3). A detailed description of the temporal segmentation algorithm is presented in the next subsection. The evaluation was separately carried out from the *Extraction,* which means that it was only focused in the effectiveness of Segmenter, as detailed in Section 3.3.2.

3.3.1 Temporal Segmentation Algorithm

The proposed segmentation algorithm uses temporal information extracted from the text, metadata and contents, in order to divide text into temporally coherent segments. These segments must be tagged with a timestamp to obtain an association between time and document terms. The length of a segment ranges from a single sentence to a multi-paragraph text. Thus, adjacent sentences with the same *chronons* must belong to the same segment. Each segment is tagged with the *chronons* found in its sentences. The document timestamp is also associated to each segment. Our approach is not yet considering the segmentation of a single sentence into small units. So, segments can have more than one *chronon*. For example, if we have two *chronons* in the same sentence, this implies a timestamp of the segment with those two *chronons*.

Figure 3.15 presents two examples with the XML tags used in temporal segmentation. In the first example, the segment *"A tempestade de* ***Domingo*** *causou alguns problemas nas redes de energia eléctrica. A empresa de energia eléctrica recebeu cerca de 31 mil chamadas."*[18] is composed of two sentences. The first one has a date (*Domingo*), which is resolved by the system of temporal information extraction. In the next sentence, the topic stays the same, so that sentence will also belong to this segment.

The segment *"Choveu na* ***sexta-feira*** *e no* ***sábado****."*[19] is other example. This segment is tagged with two *chronons*, because these two normalized dates are in the same sentence.

```
(1) <SEGMENT DN="2011-10-31">A tempestade de Domingo causou alguns
problemas nas redes de energia eléctrica. A empresa de energia
eléctrica recebeu cerca de 31 mil chamadas.</SEGMENT>

(2) <SEGMENT DN="2011-11-10 2011-11-11">Choveu na Sexta-feira e no
Sábado.</SEGMENT>
```

Figure 3.15: Examples of the temporal segmentation markers.

The flowchart of Figure 3.16 displays a high level description of the proposed algorithm.

[18] English version: **Sunday**'s storm caused some problems in electricity networks. The electricity company received about 31,000 calls.

[19] English version: It rained on **Friday** and **Saturday**.

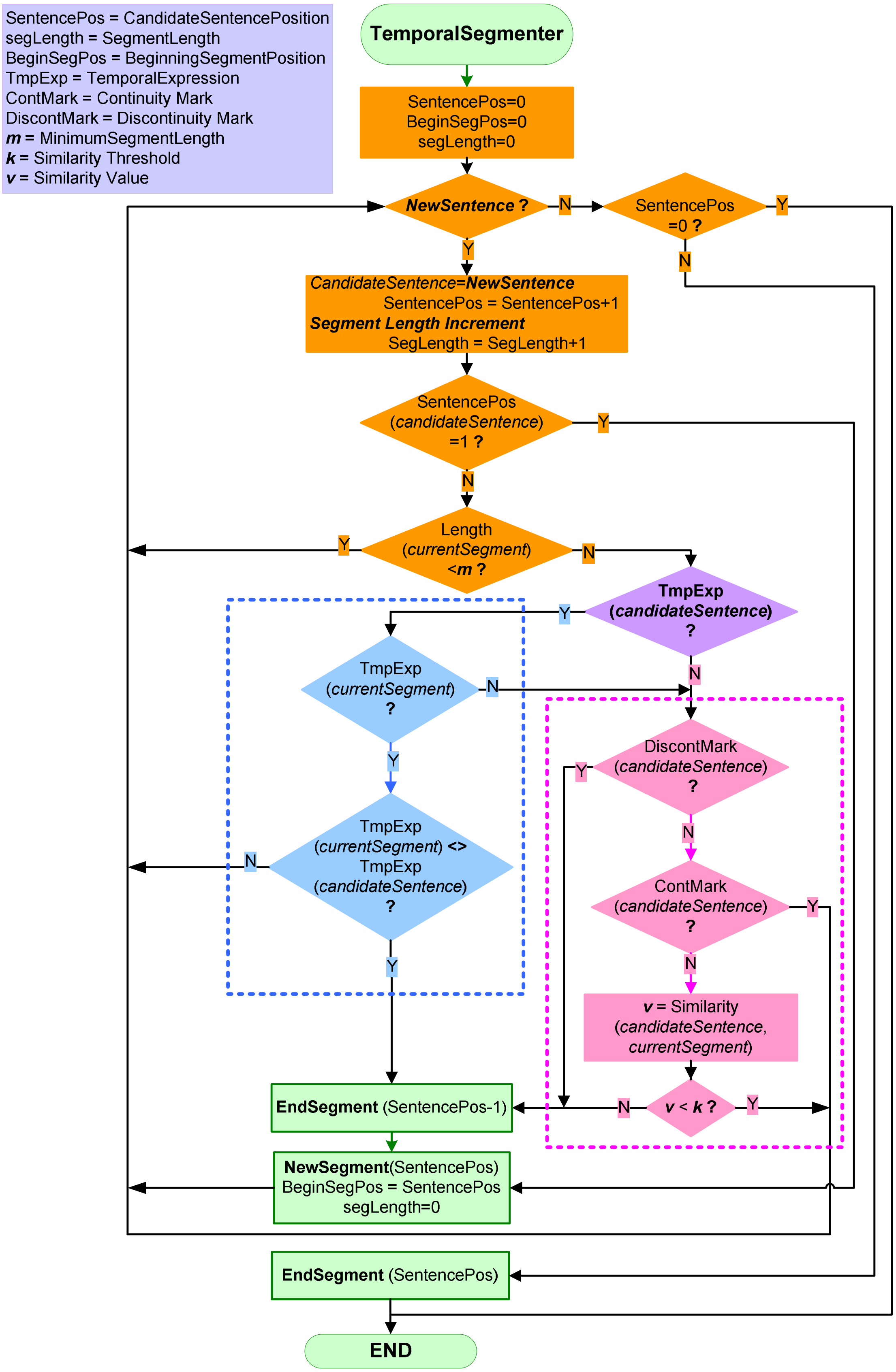

SentencePos = CandidateSentencePosition
segLength = SegmentLength
BeginSegPos = BeginningSegmentPosition
TmpExp = TemporalExpression
ContMark = Continuity Mark
DiscontMark = Discontinuity Mark
m = MinimumSegmentLength
k = Similarity Threshold
v = Similarity Value
TemporalSegmenter
SentencePos=0
BeginSegPos=0
segLength=0
NewSentence ?
N
SentencePos =0 ?
Y
Y
N
CandidateSentence=NewSentence
SentencePos = SentencePos+1
Segment Length Increment
SegLength = SegLength+1
SentencePos (candidateSentence) =1 ?
Y
N
Y
Length (currentSegment) <m ?
N
Y
TmpExp (candidateSentence) ?
N
TmpExp (currentSegment) ?
N
Y
DiscontMark (candidateSentence) ?
Y
N
TmpExp (currentSegment) <> TmpExp (candidateSentence) ?
N
ContMark (candidateSentence) ?
Y
Y
N
v = Similarity (candidateSentence, currentSegment)
EndSegment (SentencePos-1)
N
v < k ?
Y
NewSegment(SentencePos)
BeginSegPos = SentencePos
segLength=0
EndSegment (SentencePos)
END

Figure 3.16: Flowchart of the temporal segmentation algorithm.

The segmentation process begins at the sentence level. Each sentence is a candidate to start a new segment. The *candidate sentence* is compared with all text of the *current segment*, which is composed of the previous adjacent text sentence(s). There are two approaches to determine if the *candidate sentence* starts a new segment or not. These approaches are defined by a sentence analysis, based on the temporal information of the text. They are explained as follows:

1. Sentence with *chronons* (illustrated in the flowchart by the boxes in blue)**.**

If the *chronons* of the *candidate sentence* are equal to the *chronons* of the *current segment*, the sentence must belong to this segment; otherwise, this sentence starts a new segment.

2. Sentence without *chronons* (illustrated in the flowchart by the boxes in rose)**.**

(a) *Continuity or discontinuity marks.* The three first words of the *candidate sentence* are very important words because they can express a temporal discontinuity or a temporal continuity if they are temporal marks. These marks are used to express the temporal relation between successive actions or events and can signal the topic continuity or discontinuity [Bestgen and Vonk, 1995]. Table 3.13 shows some examples of these marks. The full lists, which are based on the proposal defined by Bestgen and Vonk (1995), are presented in Appendix B. Thus, if the sentence has a continuity marker, e. g. the word *"e"* (and), it remains within the *current segment*. If the marker expresses discontinuity, for example, *"depois"* (next/after), the sentence starts a new segment. Otherwise, if there is not any marker in the sentence, the sentence must be processed with the next step (2.b).

Marks	Examples (Portuguese/English)
Continuity	e/*and*, também/*also*, (n)este/*(in) this*, (n)esse/*(in) that*, eles/*they*, (…)
Discontinuity	após/*after*, antes/*before*, depois/*next*, mais tarde/*later*, então/*then*, (...)

Table 3.13: Continuity and discontinuity marks: some examples.

(b) *Similarity measure.* The procedure is based on the vector space model where the *candidate sentence* (A) and the *current segment* (B) are represented as vectors of words, A={a_1,a_2..a_n} and B={b_1,b_2..b_n}. So, the topic cohesion between the *candidate sentence* and the *current segment* is given by the cosine similarity, which is the measure of the angle between these two vectors and is computed by Formula 3.7.

$$similarity(A,B) = \cos(\theta) = \frac{\sum_i a_i b_i}{\sqrt{\sum_i a_i^2 \sum_i b_i^2}} \tag{3.7}$$

The cosine similarity is computed to determine whether to start a new segment, according to the approach used in some topic segmentation methods [Hearst, 1994]. Before calculating the similarity measure, the *candidate sentence* and the *current segment* are pre-processed to remove punctuation marks and stopwords (see the full list of stopwords in Appendix B). A threshold value for the cosine similarity is also defined to decide if the *candidate sentence* starts a new segment or not.

The computing of similarity becomes limited when using exclusively the co-occurrence of the same word, since different words are frequently used to express the same meaning in written texts. In this sense, we intent to explore in near future the using of a thesaurus and a stemming in order to improve the computing of the similarity between a sentence and a segment.

Appendix A shows a full example of the temporal segmentation by the Segmenter module.

3.3.2 Evaluation of the Segmentation Approach

The goal of this evaluation was to verify the effectiveness of the temporal segmentation algorithm in a Portuguese text collection. Indeed, the evaluation of text segmentation algorithms is a complex task. First, the selection of the reference segmentation is difficult, since the detection of the boundaries of topics involves an inherent subjectivity. Then, the choice of the measure used in the effectiveness evaluation of the algorithm, since the traditional measures, such as accuracy, precision and recall, are not the most suitable measures. Precision and recall present two main problems when used in text segmentation, as Pevzner and Hearst (2002) demonstrate in their research. The first is that when one of them is improved, the other one tends to become worse. The second is that they are not sensitive to near misses. It is common knowledge that the boundaries are not strictly defined in text segmentation. Usually, it is allowed a permissible difference in boundaries between the near sentences.

In particular, the temporal segmentation module evaluation has become an even more difficult task, besides the main reasons explained above, due to the lack of resources for text processing in the Portuguese language. The procedures used to minor all these difficulties are presented in the next section.

Experimental Setting

Time4Word is the segmentation tool developed in Perl language following the algorithm presented in Section 3.3.1. The *tool* input is an annotated and normalized corpus, a list of continuity and discontinuity marks, a stopwords list and a threshold value for the similarity measure, as shown in Figure 3.14. The corpus is in XML format, so the segments are tagged with the XML tags defined for this purpose. For example, *Time4Word* writes the tag `<SEGMENT DN="1995-04-14">` in the original text to start a new segment with the timestamp *"1995-04-14"*. The mark lists were based in the continuity and discontinuity marks defined by Bestgen and Vonk (1995). Table 3.13 shows some examples of these marks. Prepositions, conjunctions, articles and pronouns of the Portuguese language are in the *stopwords* list. A full list of marks and *stopwords* is presented in Appendix B.

An important requirement to carry out some experiments is to obtain a reference corpus composed of documents already temporally segmented. This corpus is used as a benchmark for the experiments carried out in order to determine the temporal discontinuities of the documents, following the segmentation algorithm defined.

Since there is no Portuguese collection for this purpose, we had to create a corpus composed of two subsets from HC. Next chapter presents a full description of this collection. The two subsets are called Temporal Segmentation Test Set (TSTS), used during the *Time4Word* implementation, and Temporal Segmentation Evaluation Set (TSES) that was created for the *Time4Word* evaluation. TSTS is composed of 4 documents with 82 sentences and 1,195 words, and TSES is composed of 28 documents with 401 sentences and 6,678 words. To summarize, Table 3.14 shows information about all the subsets of documents used in the evaluation.

Collection	# Collection documents	# Collection sentences	# Collection words
Temporal Segmentation Evaluation Set (TSES)	28	401	6,678
Temporal Segmentation Test Set (TSTS)	4	82	1,195

Table 3.14: Statistical information for Segmenter evaluation.

Both of the sets contains the different types of sentences that may have an influence in the determination of the segments boundaries, namely, sentences with equal and different *chronons*, sentences with continuity markers, sentences with discontinuity markers, and sentences without *chronons* and markers. The difficulty in creating large collections is related to the fact that they have to be manually segmented. In the evaluation of the temporal segmentation approach proposed by Brawsen et al. (2006), the evaluation set contained 392 sentences.

Due to the inevitable subjectivity in the definition of the topics boundaries, the selection of the reference segmentation is a difficult task. In order to solve this difficulty we created a manual segmentation corpus based on human judgments to be compared with our algorithm. Since human judges do not always agree, we measured the agreement between judges by removing the probability of chance agreement with a commonly used measure – *Kappa coefficient*.

The documents of the corpora TSTS and TSES were randomly extracted from HC and manually annotated and segmented by two human judges. The *Kappa coefficient* takes into account the fact that judges sometimes agree or disagree simply by chance. We observed an agreement of 0.91 and the *Kappa* value was equal to 0.82. Carletta (1996) states that *Kappa* greater than 0.8 signals good reliability, hence, the agreement of the judges was very good and the corpus is appropriate for the algorithm evaluation.

In addition to the traditional measure of accuracy, we also decided to use the WindowDiff (*WD*) measure. This measure, a variation of the *Pk* metric [Beeferman et al., 1999], was proposed by Pevzner and Hearst (2002) as the suitable measure for the evaluation of the text segmentation tasks. *WD* is an error metric, so the lower the value the higher the segmentation accuracy. *WD* uses a sliding window of length *k* to compare the number of expected segment boundaries to the number of experimental segment boundaries.

For the evaluation of *Time4Word* we considered not only the boundaries of the segment but also the timestamp of the segment. Due to the variation of the segment length, the width of the window (*k*) used to calculate *WD* was set to the average of the segment length in the reference segmentation using the sentence as the unit. The average of the segment length in the evaluation set is 1.18, so, *k* was set to 1. The small value for *K* is justified by the short length of the sentences and the segments. In fact, the average of the sentence length is 16 words also considering the *stopwords*.

Results

The results presented were obtained considering the boundaries of the segment and the timestamp of the segment in the evaluation sessions carried out. These results are also published [Craveiro et al., 2012].

We made several evaluations by fixing the *k* value in 1, as explained above, and by varying the threshold of the cosine similarity between 0.01 and 0.35. The graph in Figure 3.17 shows the values of *WD* obtained by varying this threshold. The worst result of the *WD*, 0.193 was obtained with a threshold of 0.25. The best result of the *WD*, 0.15 was obtained when the threshold was set to 0.04. We verified that the number of false positives increases with the increasing of the threshold.

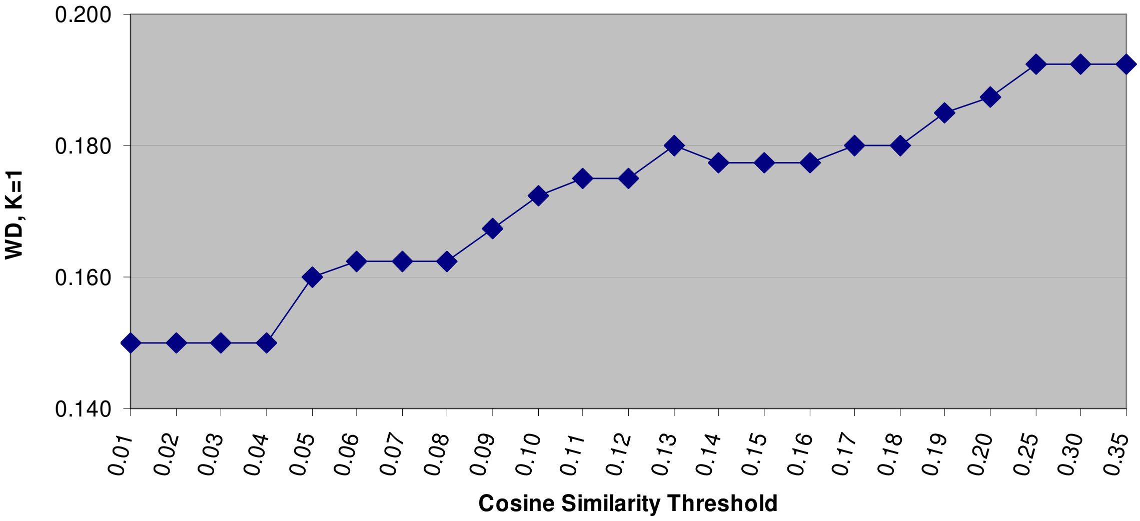


Figure 3.17: *WD*, by varying the cosine similarity threshold.

Since a timestamp of the segment can have more than one *chronon*, we calculated the agreement, the overlapping and the disagreement of the timestamps by analyzing the match between the timestamps of the segments obtained by *Time4Word* and the reference segmentation. The timestamp of two segments is in agreement when they are 100% equal. The overlapping is considered when at least one *chronon*, but not all of the segment timestamp match the timestamp of the other segment. The timestamps of the two segments are in disagreement when they are entirely different. Figure 3.18 shows some examples of segments timestamp. Example (1) shows an agreement of the two segments timestamps, since there is a match between the two timestamps. In example (2), the *chronon 2011-10-30* is not in the two timestamps, but *2011-10-31* exists in the two timestamps. So, there is an overlapping between these two segments. Since there is not any common *chronon* between the two segments of the last example, these segments are considered in disagreement.

```
(1) <SEGMENT DN="2011-10-31"> <SEGMENT DN="2011-10-31">
(2) <SEGMENT DN="2011-10-31"> <SEGMENT DN="2011-10-30 2011-10-31">
(3) <SEGMENT DN="2011-10-31"> <SEGMENT DN="2011-09-25">
```

Figure 3.18: Examples of segments timestamp.

Table 3.15 shows the minimum and the maximum value for agreement, overlapping and disagreement of the timestamps, considering the variation of the cosine similarity threshold previously discussed. Indeed, the difference between the minimum and maximum values is not very significant.

	Cosine Similarity Threshold **[0.01;0.35]**		
WD, *k=1*	**Agreement**	**Overlapping**	**Disagreement**
0.15 – 0.193	76% – 79%	1.75% –2%	19.5% – 22.5%

Table 3.15: Agreement, overlapping and disagreement of the segment timestamp.

The results obtained show a good effectiveness of *Time4Word*, even with some limitations. Although the accuracy was about 78%, the *WD* was not so penalized – *WD=0.15*. This means that some incorrect boundaries of the segment are within the *k-sentence* window used by *WD* metric. We verified that the incorrect boundaries of the segments have been determined in a particular case – sentences without dates and where the similarity calculation was applied. Seeing that the similarity measure only takes into consideration the occurrence of the same word, and in writing it is frequent to resort to words with the same or nearly the same meaning to express the same idea, the use of synonyms and a stemmer will certainly improve the results.

3.4 Summary

This chapter focused on temporal retrieval preprocessing required in order to incorporate time dimension in retrieval models. In our work, besides the temporal information extraction from documents and queries, this preprocessing also includes the temporal segmentation of the texts for obtaining the temporal relationships between words. The segmentation divides the text using the temporal discontinuities, which are detected by the temporal information found in documents, namely metadata and content. Based on the two main tasks of the retrieval preprocessing, we propose the extraction and the segmentation approaches.

The extraction approach is to obtain a normalized format of the temporal expressions identified in Portuguese texts, following a sentence-by-sentence processing. Note that this approach distinguishes from other works because it does not require a linguistic analysis of the text. A set of temporal expressions patterns are used to identify and classify the temporal expressions. We also propose an approach to create these patterns, using the aggregation of temporal expressions.

The segmentation approach uses the information given by the extraction to find the temporal discontinuities of Portuguese texts, dividing the text into segments. Words in the same segment share the same temporality.

The evaluation was only focused in the effectiveness of the two approaches. Each approach was separately evaluated from the other one. Even, with some limitations, both approaches achieved promising results, as presented in this chapter. Some improvements are required in the extraction, for instance, resolving all the indexical references. The segmentation must also be improved, namely in the process of topic change detection when using, for instance, a thesaurus and a stemming.

We developed a toolset based on the two approaches, which is composed of the *Extraction*, and the *Segmentation* tools. All the modules of the tools are also described in this chapter.

Chapter 4

Temporal Characterization of the Text Collections

This chapter describes the two collections, Second HAREM and CHAVE, used in the various evaluation sessions of all the software developed. A temporal characterization based on a statistical analysis was carried out for these collections, in order to understand how temporal information appears in the documents. The knowledge obtained was applied in the document model definition, which is presented in the next chapter. The temporal characterizations are detailed in this chapter, where the metrics used and the results obtained are also fully explained.

4.1 Introduction

The main objective of this work is to enhance the information retrieval task, namely incorporating temporal attributes in the retrieval process. A key idea underlying our approach is to divide the text into segments, thereby establishing a relationship between words and time information recognized in the document content. This relationship is included in the information retrieval model to improve the result lists.

The analysis of the text collections presented in this chapter was carried out to better understand the behavior of the temporal information so as to provide a temporal characterization of Portuguese documents which has a crucial role in the two main facets of this work: the temporal document segmentation and the definition of a time-aware information retrieval model. The Gold Collection, a subset of the Second HAREM collection, and the CHAVE collection were the collections of Portuguese documents used in this temporal characterization. These two collections are available at Linguateca site [lin, nd].

The temporal characterization of CHAVE considered not only the documents, but also, the topics and their relevant documents. A further analysis was performed to validate the usability of this collection for temporal *ad hoc* retrieval research and to understand the time sensitivity of topics.

4.2 Second HAREM Collection

The Second HAREM Collection was created by Linguateca for the second evaluation contest for named entity recognition (NER) in collections of documents in Portuguese.

The texts of the collection are only written in Portuguese but in two variants of the language: European Portuguese and Brazilian Portuguese. The documents are obtained from various sources, such as the Portuguese newspaper *Expresso*[20], Wikipedia, “Brasil Cultura” website, etc. and are structured in different genres – journalistic, educational, blog, questions, FAQ, dialog, literary, juridical and advertising. This collection has a total of 1040 documents with 33,712 sentences and a total of 668,817 words. A subset of this collection, named as Gold Collection, was also defined by Linguateca in order to evaluate the identification and the classification tasks of the Second

[20] http://www.expresso.pt [September 20th, 2015]

HAREM. The particularity of the Gold Collection is that its documents were manually annotated using time HAREM guidelines [Hagège et al., 2008a]. The Gold Collection is composed of 129 documents with 4,465 sentences and a total of 74,350 words. The Second HAREM collection and the Gold collection are available at HAREM site[21] and are properly detailed by Mota et al. (2008a).

The Gold Collection is also composed of text structured in diverse literary genres, such as the main collection Second HAREM Collection; news article is the predominant literary genre with 35%. Different genres can probably make a better understanding of how time related expressions appear in its content, since, for example, the temporal content of news article might be different from a private blog.

Each HAREM collection is available in a single XML file and a document is marked up with DOC, the XML element, and DOCID, the XML attribute, which is the document identification. Each paragraph of the document is marked up with P, a XML element, also named tag. Figure 4.1 shows an example of a document extracted from the collection (see the complete document in Appendix C).

```
<DOC DOCID="hub-28874">
<P>H5N1: Mais de 32 mil mortos se pandemia atingisse Portugal</P>
<P>Mais de 32 mil pessoas poderiam morrer se uma pandemia de gripe humana de
origem aviária atingisse Portugal, (...).</P>
<P>Na quinta-feira, a Organização Mundial de Saúde (...).</P>
(...)
</DOC>
```

English version

```
<DOC DOCID="hub-28874">
<P> H5N1: More than 32 thousand deaths if pandemic hits Portugal</P>
<P>More than 32 thousand people could die if a human influenza pandemic of
avian origin reaches Portugal, (...).</P>
<P>On Thursday, the World Health Organization (...).</P>
(...)
</DOC>
```

Figure 4.1: Document *hub-28874* from the Second HAREM collection.

4.2.1 Temporal Characterization

The temporal characterization carried out with the Gold Collection had an important role in the definition of the algorithm to create the temporal segments, detailed in the previous chapter. To define our document model, we consider at the same level of importance both the attributes used by information retrieval models, such as document representative keywords and their positions, and

[21] http://www.linguateca.pt/HAREM [September 20th, 2015]

the following attributes: document timestamp, *chronons* and their positions, and the beginning and the end of temporal segments.

These attributes support the temporal characterization presented in this section by using two types of measures: temporal measures that describe the temporal content of the documents, and segment measures that can give some evidence on how to segment the document and the dimension of segments, such as the distance between *chronons*, the position and frequency of the temporal information inside a sentence, paragraph or document.

We also consider some of the temporal measures proposed by Alonso (2008), such as temporal richness, temporal specificity and temporal boundaries of documents. The temporal richness is defined as the percentage of *chronons* in the document content relative to the total number of *chronons* in the collection. The temporal specificity of a document is the most frequent granularity of its dates. The temporal scope can be given by the temporal boundaries. Table 4.1 presents all the measures used for the temporal characterization of documents.

Segment Measures	Distance between *chronons* Temporal position inside a sentence, paragraph or document Temporal frequency inside a sentence, paragraph or document
Temporal Measures	Temporal richness Temporal specificity Temporal boundaries Temporal scope

Table 4.1: Measures used in temporal characterization.

Thus, the temporal analysis is focused on temporal content of documents given by *chronons*. The Gold Collection does not include the *chronons*, since it is temporally classified but not normalized. For this reason, the collection was submitted to the Resolver module (see Section 3.2.3). The output of this module was the Gold Collection, duly classified and normalized with the *chronons* (henceforth NG Collection), which was the collection used in this analysis. Note that, since the Resolver module does not have a 100% effectiveness, the results presented here are subject to some errors.

The NG Collection is composed of 129 documents with 4,465 sentences and 74,350 words. From the 1192 temporal expressions classified, 870 *chronons* were obtained in the NG Collection. There is temporal information in about 98% of documents, though 95% of documents have at least one *chronon*. The number of *chronons* could be higher, since 45% of the documents have not the document timestamp, which is very important to resolve some references. Figure 4.2 shows a histogram of document occurrences, with the number of temporal expressions and *chronons* found per document.

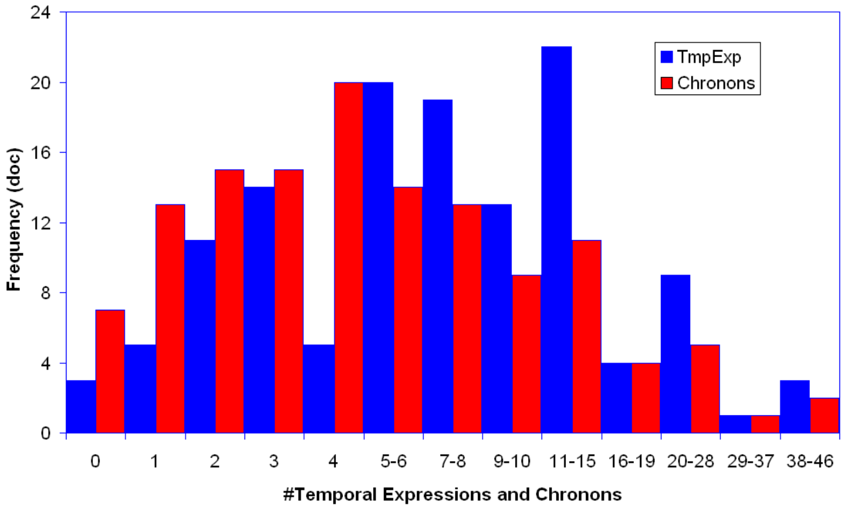


Figure 4.2: Frequency of temporal expressions and *chronons* per document.

The temporal content of the documents is given by *chronons* that are associated to the finest timeline granularity. We verified that the temporal specificity of the collection is *year*. However, considering this measure per document, *day* is the temporal specificity for a considerable number of documents (about 49%), while *year* is for 44% of the documents, as shown in Figure 4.3.

We also analyzed this measure according to the literary genres, since the temporal information of a document might give some clues about which is its literary genre, such as, news article, narrative text, etc. We verified that *year* is the temporal specificity of essays, didactic and questions text. Blogs, legislative text and news article have *day* as the temporal specificity. The temporal specificity of news article is not well defined, since there is a small difference of three *chronons* between *day* (42.3%) and *year* (41.2%).

After the analysis of temporal specificity, the obvious conclusion was that *year* and *day* are the most representative time granularity of the NG collection.

The higher temporal richness was about 4.5%, shared by two documents. This means that these two documents have approximately 9% of the *chronons* in the collection.

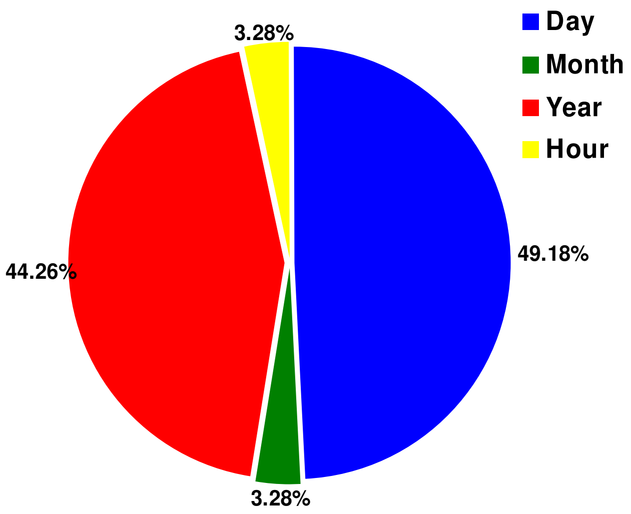


Figure 4.3: Temporal specificity per document.

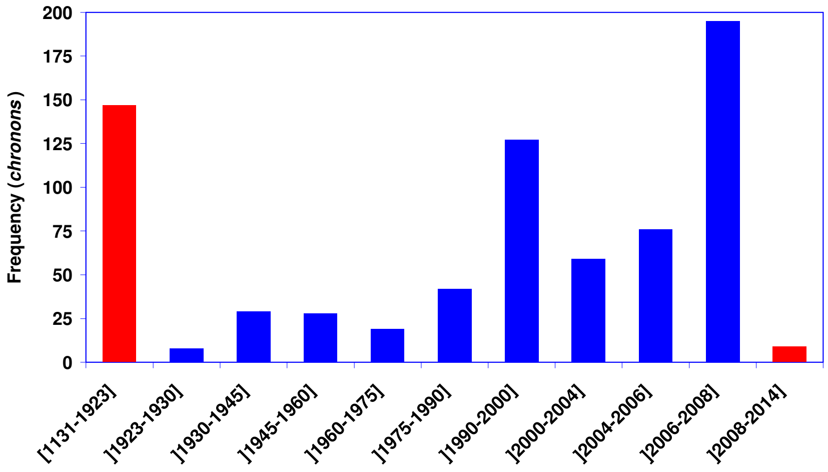


Figure 4.4: Temporal scope of the NG collection.

The documents of the NG Collection are dated from 1924 to 2008, but their temporal scope is higher, ranging from 1131 to 2014. The histogram of Figure 4.4 illustrates the temporal scope of the collection, considering the *year* as the temporal specificity. The blue bars represent the *chronons* in the range of the documents timestamps. The *chronons* outside this range are displayed as red bars. Furthermore, we verified that didactic texts are the ones most responsible for a narrative of past events.

The number of *chronons* in a document can give the potential number of segments expected in the document. As Table 4.2 reports, the average in the collection is of about 6 segments per document and the minimum value is 1, which means that if there is no temporal information in the document content, the only segment of the document is labeled with the document timestamp.

Considering that the sequence of words, sentences, and paragraphs after or before a temporal reference refers events at a given time, relevant information is the number of words between sequences of two *chronons*, which also gives the density of *chronons* in a document. Table 4.3 reports the distance between two *chronons* measured by the number of paragraphs, sentences and words. An average of about 4 sentences is the value obtained for the segment length.

	Maximum	**Minimum**	**Average**	**Median**
Temporal Expressions	46	1	9.23	7
Chronons	39	1	6.74	4

Table 4.2: Temporal expressions and *chronons* per document.

Distance	**Maximum**	**Minimum**	**Average**	**Median**
in paragraphs	50	0	2.36	1
in sentences	68	0	4.47	2
in words	927	1	71.19	36

Table 4.3: Distance between *chronons*.

The frequency of the *chronons* inside paragraphs and sentences is also important to determine the unit to be employed in the division of the document. The maximum number of *chronons* found in a document was 9 and 8 in paragraph and sentence, respectively. However, the graph in Figure 4.5 shows that the higher frequency obtained was 1 *chronon* per paragraph and the same value per sentence. The median value is exactly 1 and the average is a little bit higher, with 1.46 and 1.21 to paragraph and sentence, respectively. This means that the sentence would be a good choice for the division unit of the document.

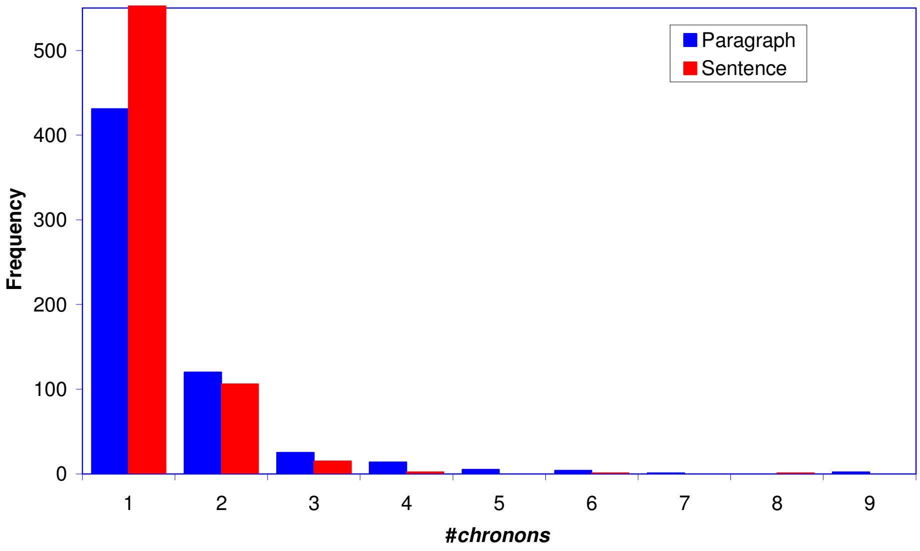


Figure 4.5: Number of *chronons* per paragraph and per sentence.

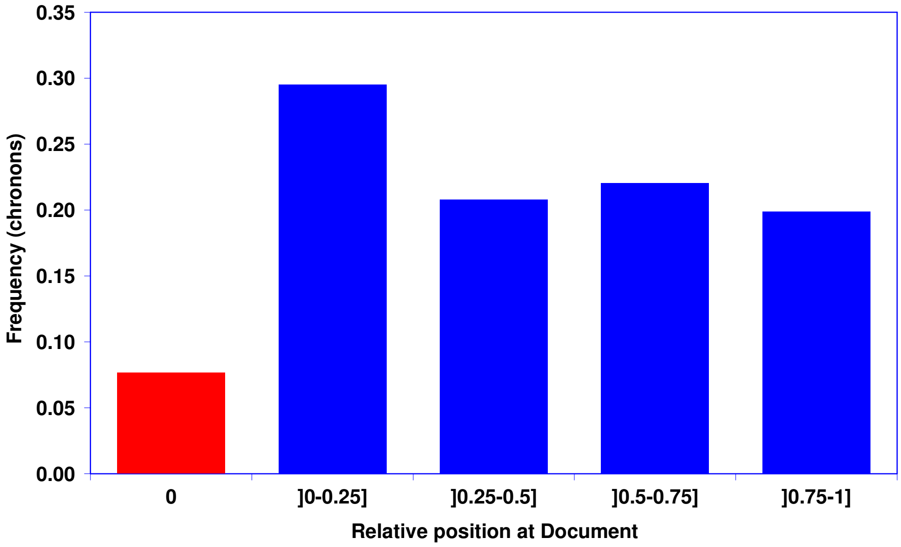


Figure 4.6: *Chronons* position at documents.

Figure 4.6 shows that the *chronons* occur more frequently in the first quarter of the documents. This means that this is the temporally richest part of documents; therefore, this is a candidate part to have more text segmentation. In x-axis, this figure represents the relative position of *chronons* at documents. The timestamp of the documents are represented at position 0.

4.3 CHAVE Collection

The CHAVE collection is the only collection available with documents in Portuguese and all the resources required can be used by information retrieval systems. This collection was created by Linguateca for the participation of the Portuguese language in three tracks of the CLEF[22] Initiative (Conference and Labs of the Evaluation Forum, formerly known as Cross-Language Evaluation Forum), namely, the *ad hoc* IR, the Question Answering (QA) and the GeoCLEF. This collection has been updated every year since 2004 as a result of Linguateca participation in the organization of CLEF, which has been very important to enhance the participation of the community involved in Portuguese language processing in this international evaluation contest to promote and make available public resources.

The CHAVE collection is composed of full-text from two major daily Portuguese and Brazilian newspapers, namely the PUBLICO[23] and the Folha de São Paulo[24], from complete editions of 1994 and 1995. Although CHAVE is a monolingual collection, the texts are written in two variants of the language: European Portuguese and Brazilian Portuguese. This collection has a total of 210,734 documents with 4,682,363 sentences and a total of 90,646,837 words.

Table 4.4 presents some quantitative information of the PUBLICO and Folha de São Paulo Collections [lin, nd]. The Portuguese newspaper PUBLICO has fewer editions because it is neither published on Christmas Day nor on New Years Day.

					Words	
Collection	**Newspaper**	**Editions**	**Documents**	**Size**	**Total**	**Distinct**
	1994-1995	(daily)		(KBytes)		
CHAVEPublico	PUBLICO	726	106821	348078	54947072	472817
CHAVEFolha	Folha de São Paulo	730	103913	226690	35699765	393885

Table 4.4: Information about the *PUBLICO* and the *Folha de São Paulo* collections.

[22] http://www.clef-campaign.org [September 20th, 2015]
[23] http://www.publico.pt [September 20th, 2015]
[24] http://www.folha.com.br [September 20th, 2015]

The CHAVE collection is composed of several SGML files. Each of these files stores all the articles published in a day. A CHAVE document contains a single article from the newspaper, and some metadata information. An example is displayed by Figure 4.7. This figure only shows a part of the document PUBLICO-19940127-135, but the complete document is displayed in Appendix C.

```
<DOC>
<DOCNO>PUBLICO-19940127-135</DOCNO>
<DOCID>PUBLICO-19940127-135</DOCID>
<DATE>19940127</DATE>
<CATEGORY>Desporto</CATEGORY>
<AUTHOR>JMF</AUTHOR>
<TEXT>
BMW 325 tds
Você pediu diesel?
Ai, ai: por vezes há automóveis de que nos custa separar (...)
</TEXT>
</DOC>
```

English Version

```
<DOC>
<DOCNO>PUBLICO-19940127-135</DOCNO>
<DOCID>PUBLICO-19940127-135</DOCID>
<DATE>19940127</DATE>
<CATEGORY>Sport</CATEGORY>
<AUTHOR>JMF</AUTHOR>
<TEXT>
BMW 325 tds
Did you ask for diesel?
Oh, Oh: sometimes it is not easy to leave some cars (...)
</TEXT>
</DOC>
```

Figure 4.7: Document *PUBLICO-19940127-135* of the CHAVE collection.

The test questions, also known as topics, and their relevance judgments are also made available by Linguateca. The details of the creation of this collection were published by Santos and Rocha (2004, 2005). For the purpose of this work, the suitable resources are the resources available to the *ad hoc* IR track. Note that the evaluation of information retrieval systems requires documents, topics and relevance judgments. Topics are the queries to be submitted to system. Relevance judgments are an important part of the collection. They contain the information about the relevance of documents for each topic. The information about the resources used in this analysis is presented in Table 4.5.

CLEF edition	**Topics**	**Newspaper** (1994-1995)
CLEF 2004	C201 – C250	PUBLICO
CLEF 2005	C251 – C300	PUBLICO, Folha de São Paulo
CLEF 2006	C301 – C350	PUBLICO, Folha de São Paulo

Table 4.5: CHAVE collection: resources used in this work.

All the resources are in TREC format, so the documents and the topics are annotated in SGML. The document structure is defined by *SGML Document Type Definitions* (DTDs). The document structure is marked up with the following elements: document identification (DOCNO and DOCID), publication date (DATE), literary genre (CATEGORY), author's name (AUTHOR) and the full text of the article (TEXT). The titles are not marked, so they are also included in TEXT, appearing as free text. CATEGORY and AUTHOR are optional elements, so they do not occur in all documents. For example, CATEGORY does not exist in CHAVEFolha (1994), and CHAVEFolha (1994-1995) does not have the AUTHOR element.

The queries used in experiments on information retrieval systems are generated from the collection topics. The topic structure is composed of the following elements: topic identification (*num*), a few words which better describe the topic (*PT-title*), one sentence which describes the topic subject (*PT-desc*), and a more detailed topic description, concisely indicating what makes a document relevant to the topic (*PT-narr*). Figure 4.8 shows the topic 251 as example.

```
<top>
<num> C251 </num>
<PT-title> Medicina alternativa </PT-title>
<PT-desc> Encontrar documentos sobre tratamentos (...). </PT-desc>
<PT-narr> Documentos relevantes devem fornecer (...). </PT-narr>
</top>
```

English Version

```
<top>
<num> C251 </num>
<PT-title> Alternative medicine </PT-title>
<PT-desc> Find documents about treatments (...). </PT-desc>
<PT-narr> Relevant documents should provide (...). </PT-narr>
</top>
```

Figure 4.8: Topic 251 of the CHAVE collection.

Relevance judgments are very important as they provide the way to carry out the effectiveness of the evaluation of the information retrieval systems. The format of the relevance judgments is composed of the *topic number*, the *feedback iteration*, the *document identification* (DOCNO element) and the *relevancy* which is represented as a binary code where zero (0) is not relevant and one (1) is relevant. Therefore, the document position in the file does not indicate a greater or lesser relevancy degree.

Since a test collection has thousands of documents, it is not always possible to perform an analysis of the relevance of all existing documents for each topic. Therefore, the relevance judgments files only contain the documents judged by the human assessors, which are people that determine manually the documents relevance for each topic. The other documents are not considered in the evaluations. A document is classified relevant by the human assessor if any piece of the document is relevant for the topic judged, irrespective of its length. See an example of the topic C251 relevance judgments in Table 4.6. The document FSP940130-099 was considered relevant for this topic, while the other two documents were considered non-relevant. Note that the *feedback iteration* is not used in relevance judgments of CHAVE, therefore its value is always 0.

Topic	ITERATION	DOCNO	RELEVANCY
C251	0	FSP940128-148	0
C251	0	FSP940130-099	1
C251	0	FSP940130-100	0

Table 4.6: An example of the CHAVE relevance judgments.

For this collection, we verified that all the topics were judged, although 4 topics, namely C216, C220, C227, and C240, do not have any relevant document judged. The number of relevant and non-relevant documents for each topic judged is very different (see Table D.1 in Appendix D). Note that, there are much more documents classified as non-relevant, than relevant. On average, each topic has 41.7 relevant documents, for 378.3 documents classified as non-relevant. Besides the average, Table 4.7 shows the median value that is close to the average for non-relevant documents. However, this value is lower than the average for relevant documents, which means that there are more topics with a value lower than 41 relevant documents. The maximum and minimum are also presented in this table.

	Number of Documents Judged			
	Maximum	Minimum	Average	Median
NonRelevant	939	90	378.3	372
Relevant	266	0	41.7	28.5

Table 4.7: Statistics about non-relevant and relevant documents judged.

4.3.1 Temporal Characterization

We carried out the temporal characterization of the CHAVE collection. It helps us in understanding the relationship between the distributions along timelines of relevant and non-relevant documents, for a given topic. Furthermore, this characterization gives the temporal sensibility for each topic.

Although the timestamp of the documents are considered in the temporal characterization, the focus is the temporal content of documents that can be later used in information retrieval models; in other words, the temporal information represented as *chronons*.

So, first, the CHAVE collection was processed by our testbed system, from the Annotator module until the Segmenter module. Then, the measures used in this temporal characterization were computed. Note that the results presented here are subject to an error for two main reasons: first, our system does not have an effectiveness of 100%, as we explained in the last chapters; second, the original documents have also some spelling mistakes in their contents, for example, *"Em Março de **9160**[25]"*. However, looking at the text, this expression must be *"Em Março de **1960**"*.

So as to obtain the maximum information of temporal content of the collection, we did not limit the temporal analysis to documents, but the topics and their relevant documents were also analyzed. In this subsection, the analysis peformed is presented following the order: collection, documents, topics, and relevant documents of topics.

By the results obtained, published by Craveiro et al. (2015) and presented below, the temporal characterization of CHAVE shows that the collection is sufficiently rich to be used in temporal information retrieval evaluation.

Collection

The temporal scope of the CHAVE collection is from year *94AD* to *2577*, though their documents are dated in [*1994-01-01; 1995-12-31*]. Temporal information was found in the content of about 90% of documents, but as this information is not fully resolved, the percentage of documents with at least one *chronon* is lower (86%). A total of 869,051 *chronons*, but only 12,796 distinct *chronons*, were obtained from the 1,022,846 temporal expressions which were identified and classified. Note that these two values do not have a direct correspondence, due to some temporal expressions do not have a *chronon*, but others have two *chronons*. For example, the temporal expression "*1992-93*" is represented with two *chronons*: *1992-XX-XX* and *1993-XX-XX*.

DATE was the classification given to about 90% of the temporal expressions. The graph in Figure 4.9 shows the frequency of *chronons* in their direct corresponding timelines. For example, *1995-09-XX* is associated to the timeline with month granularity. We observed that the *chronons* are distributed by four timelines: *hour*, *day*, *month*, and *year*. As shown by this graph, the temporal

[25] English version: *In March of **91**60 (1960)*

specificity of the CHAVE collection is *day*. Indeed, this situation was already expected, since documents are news of daily newspapers.

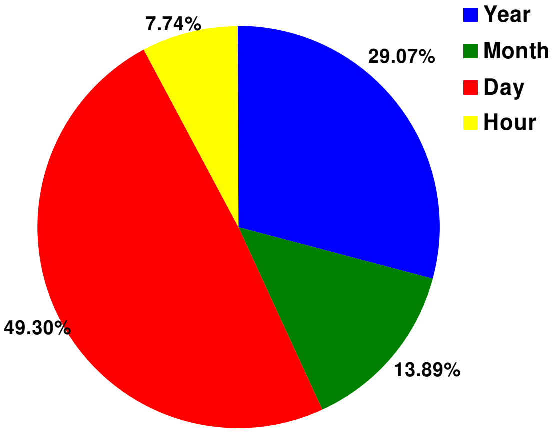


Figure 4.9: *Chronons* per timeline, T={*Th*, *Td*, *Tm*, *Ty*}.

It was not possible to carry out any temporal segmentation in 29,847 documents (14%), due to the fact that these documents do not have any *chronon* (see Figure 4.10). In this case, the document has a single temporal segment.

A total of 1,255,416 temporal segments were found in the collection with an average of about 69 words and 1 *chronon* per segment. The number of words and distinct words is very close, which means that in most segments, there are few repeated words. Table 4.8 shows the maximum, minimum, average, and median values of the quantity of *chronons*, words, and distinct words, per temporal segment.

	Maximum	Minimum	Average	Median
#*Chronons*	29	0	0.665	1
#words	4202	1	68.64	34
#DistinctWords	1305	1	53.67	33

Table 4.8: Some statistics per temporal segment.

Documents

The results show that about 54% of the documents have at least 1 *chronon* and a maximum of 4 *chronons*, as represented by the graph in Figure 4.10. This graph displays the percentage of documents with a given number of *chronons*, which varies between 0 and the maximum value, 228.

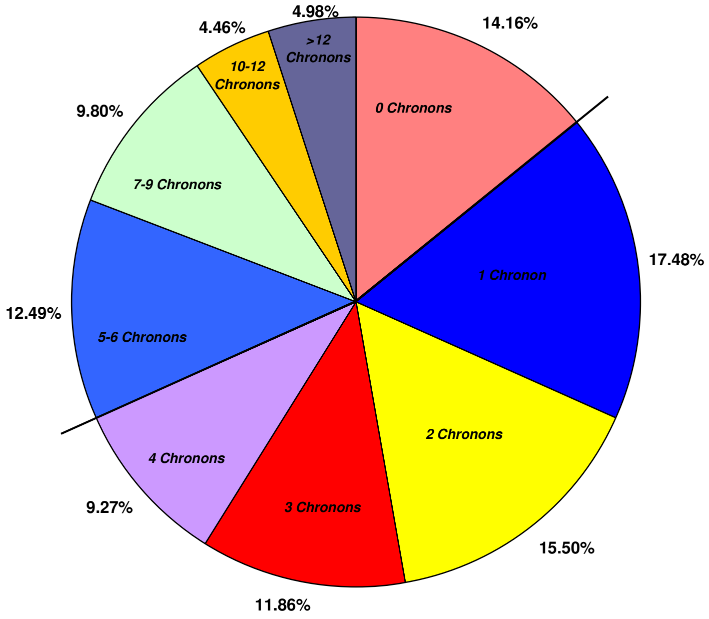


Figure 4.10: Frequency of *chronons* per document.

The number of temporal segments per document closely follows the number of *chronons* of a document, as foreseen, since the *chronons* give an indication about the number of temporal discontinuities per document. Thus, the average number of temporal segments per document is about 6 with a median value of 5 and a range from 1 to 388.

The highest temporal richness (about 0.0003%) is shared by two documents, each one with 228 *chronons*, the maximum number of *chronons* per document.

Figure 4.11 shows the percentage of documents with a temporal specificity of year, month, day or hour. Although *day* is the predominant temporal specificity (about 47% of the documents), there is also a considerable number of documents (about 40%) with *year* as the most representative time granularity.

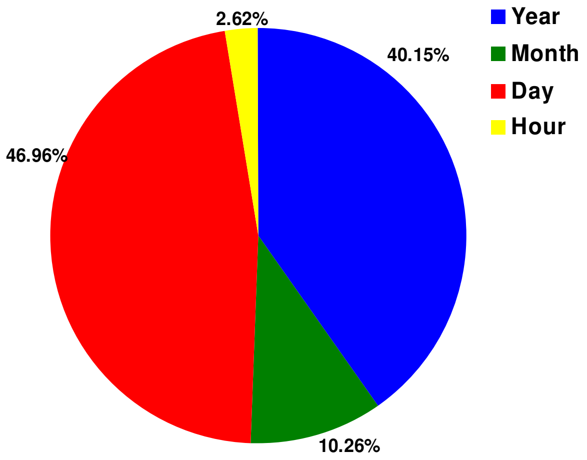


Figure 4.11: Documents per temporal specificity.

As displayed in Figure 4.12, the *chronons* that have an occurrence in more documents are very close to the publication date of the collection documents, represented with bars in red. This figure shows the 6 *chronons* that occur in more documents.

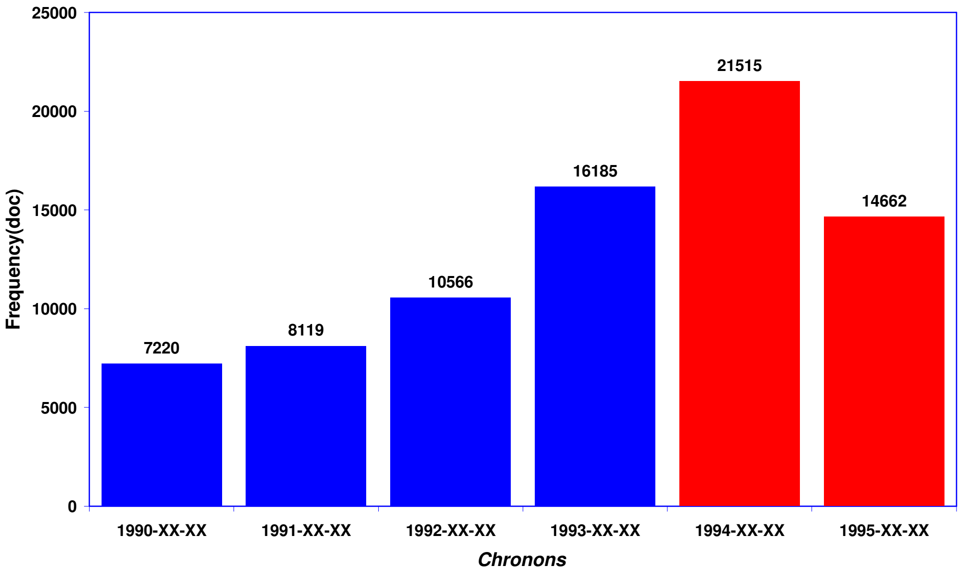


Figure 4.12: Number of documents per *chronon*.

### Topics

The topics of CHAVE are grouped into three groups: (1) C201-C250, (2) C251-C300, and (3) C301-C350, as showed in Table 4.5. We observed that the second group only has one topic with temporal information, but there is not any *chronon*, since the only temporal expression *"nos dias de hoje"*[26] has a classification of GENERICO, which means that this temporal information cannot be used as explicit temporal information. Table D.2 in Appendix D displays all the temporal information found in the collection topics.

Table 4.9 shows some information about the temporal expressions and *chronons* of the topics. Almost all the temporal expressions in the text of the topics could be mapped into *chronons*, since these expressions were classified as DATE, having explicit temporal references.

We verified that only 32 of the 150 topics (21.3%) have explicit temporal information. Almost all these topics aim to obtain documents referencing to events that occur during the time period of the collection (1994-1995), only considering the publication date of the documents. More precisely, the temporal information of 13 topics of group 1, and 17 topics of group 3 reference dates in that time period. Only two topics, C339 and C221, have references which are outside of that time period. The topic C339 demands information about a past event, which happened in *December 1993*. The topic C221 is also the only topic that requires documents referring to a future event which occurred in the year of *2002*.

				#Topics		
	#Temporal Expressions	***#Chronons***	**Temporal Specificity**	***Explicit* TempInf**	**Past Events**	**Future Events**
(1) TopicsC201-C250	24	22	Y	14	0	1
(2) TopicsC251-C300	1	0	*N.A.*	0	0	0
(3) TopicsC301-C350	25	26	Y	18	1	0

Table 4.9: Temporal information in the CHAVE topics.

Topics and their relevant documents

Then, we present an analysis of the relationship between the topics and their documents judged as relevant documents. As CHAVE has 4 topics without any relevant document judged, for the following results these topics were not considered. All these topics are in the first group ((1) TopicsC201-C250).

[26] English version is: "*these days*"

The number of *chronons* and distinct *chronons* found in the relevant documents are displayed in Table 4.10. These values were calculated taking into account the three groups of topics. This table also shows the number of relevant documents with and without *chronons* in their content. We verified that a few number of relevant documents do not have any *chronon*. About 93% of relevant documents have at least one *chronon* in their content, which means that temporal information is present and it can be used in this collection.

	Relevant Documents		**#Relevant Documents**	
	#*Chronons*	**#Distinct *Chronons***	**WITH *chronons***	**WITHOUT *chronons***
(1) TopicsC201-C250	3,895	1,799	627	51
(2) TopicsC251-C300	17,501	7,723	2,714	190
(3) TopicsC301-C350	18,606	6,510	2,492	185

Table 4.10: Temporal information of relevant documents by topic groups.

As shown by Figure 4.13 the temporal information has a stronger presence in the relevant documents of CHAVE. Only 2 topics have at least one *chronon* in only about 50% of their relevant documents. About 6% of the 144 topics have temporal information in a percentage of 49%-80% of their relevant documents. About 60% of the topics have *chronons* in more than 95% of their relevant documents.

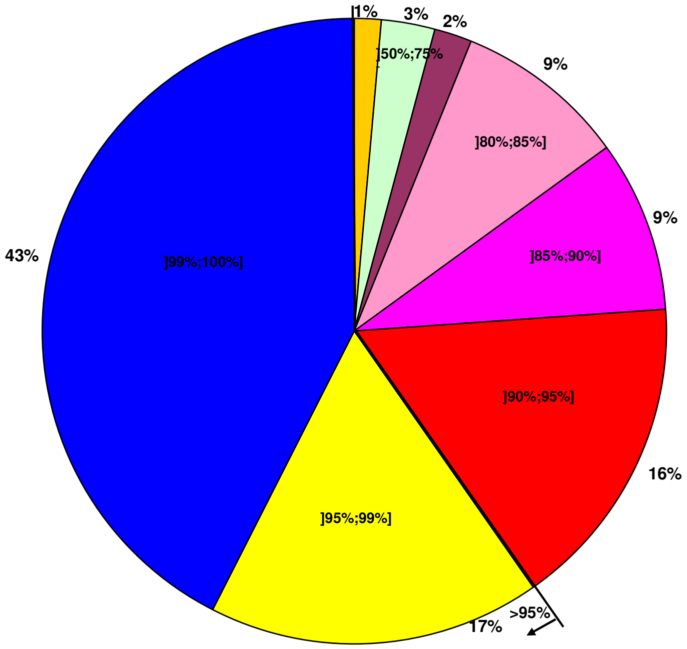


Figure 4.13: Percentage of topics with temporal information in the set of relevant documents.

Table 4.11 shows the maximum, minimum, average and median values of *chronons* and distinct *chronons* found in the relevant documents for each topic. We verified that the values obtained for the topics of group 2 are near the values of the topics of group 3. Although, the maximum value of *chronons* that exists in the set of relevant documents of a topic in group 3 is greater than the same value of group 2, the average and the median values are very close. In general, the topics of group 1 have less relevant documents than the topics of other groups, as shown by Table D.1 in Appendix D. So, it is normal that the topics of this group also have less *chronons*. However, the three groups of topics have approximately the same percentage of distinct *chronons*. We can notice that the distinct *chronons* are around 30% of all *chronons* for each group of topics. Group 1 has 35%, group 2 has 33% and group 3 has 27% of distinct *chronons*.

		Maximum	**Minimum**	**Average**	**Median**
(1) TopicsC201-C250	*Chronons*	842	1	84.67	37.5
	Distinct Chronons	293	1	39.11	20.0
(2) TopicsC251-C300	*Chronons*	1863	12	350.02	246.5
	Distinct Chronons	612	8	154.46	145.5
(3) TopicsC301-C350	*Chronons*	2226	8	372.12	241.5
	Distinct Chronons	593	8	130.20	104.5

Table 4.11: Statistics of relevant documents per CHAVE topic.

Figure 4.14 displays two timelines T_y and T_m with the representation of the *chronons* found in the text of the CHAVE topics. These *chronons* are represented in their correspondent timeline. Note that the *chronons* with a granularity of month are only marked in timeline T_m and are not mapped to the timeline T_y. There is not any *chronon* which represents a complete date with year, month and day. Only 8 of 32 *chronons* have representation in the timeline with month granularity; the others are not so specific, having only the year. As previously referred, almost all topics request information in the interval of time publication of the CHAVE documents; there is only one topic (C221) with a date after this interval and other (C339) with a date before the same interval.

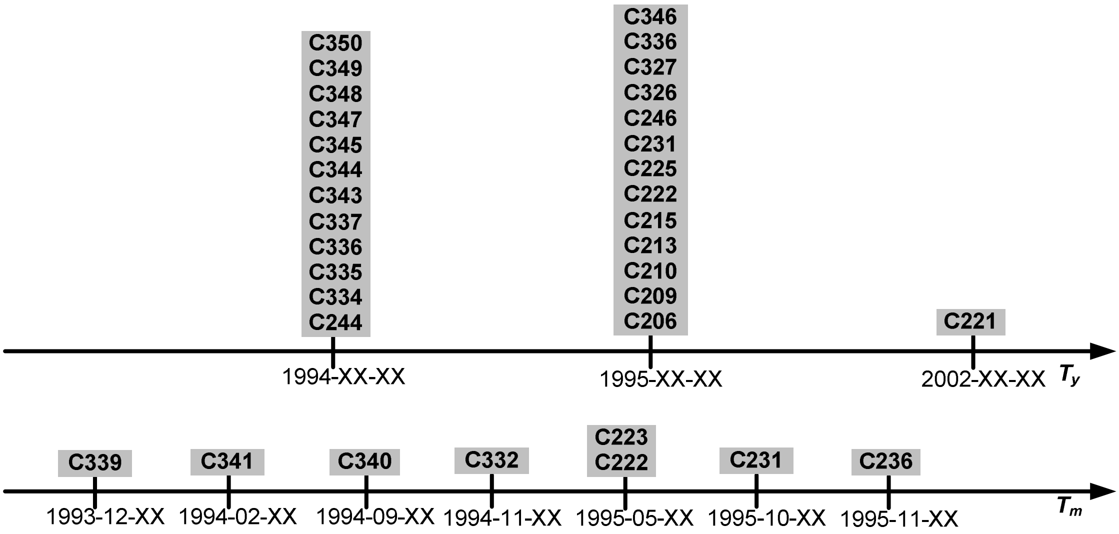


Figure 4.14: *Chronons* of the CHAVE topics represented in the timelines T_y and T_m.

We also analyzed the relationship between the *chronons* of the topic and the *chronons* found in their relevant documents. For each topic, we computed the number of distinct *chronons* of their relevant documents that are equal when they are mapped into the same timeline of the topic *chronons*. The number of distinct *chronons* of the relevant documents which represents a date before or after the date of the topic is also computed.

Figure 4.15 and Figure 4.16 show 3 values computed for each topic whose *chronons* are mapped to the timeline T_y, and T_m, respectively. The bars in blue represent the quantity of *chronons* equal in topic and relevant documents. The number of *chronons* of relevant documents which are before the *chronon* of the topic is displayed by bars in red. The other value is represented by bars in yellow.

We verified that there are also temporal references of about 60% of topics, represented in Figure 4.15, in the relevant documents of the topic. Note that the content of the relevant documents has more temporal references to present or past events than future events. In general, the temporal references, both in topic text and content of relevant documents, are in the interval of date publication of the CHAVE documents.

The only two topics that have a date outside of the time interval of CHAVE are C221 and C339. The topic C221 requires information about a future event. For that reason, this topic has more *chronons* of the content of relevant documents which are previous to the topic *chronon 2002-XX-XX*. The *chronon* of the other topic is *1993-12-XX*, a date before the time interval of CHAVE documents, therefore there are much more *chronons* of the relevant documents, which are dated after *1993-12-XX*.

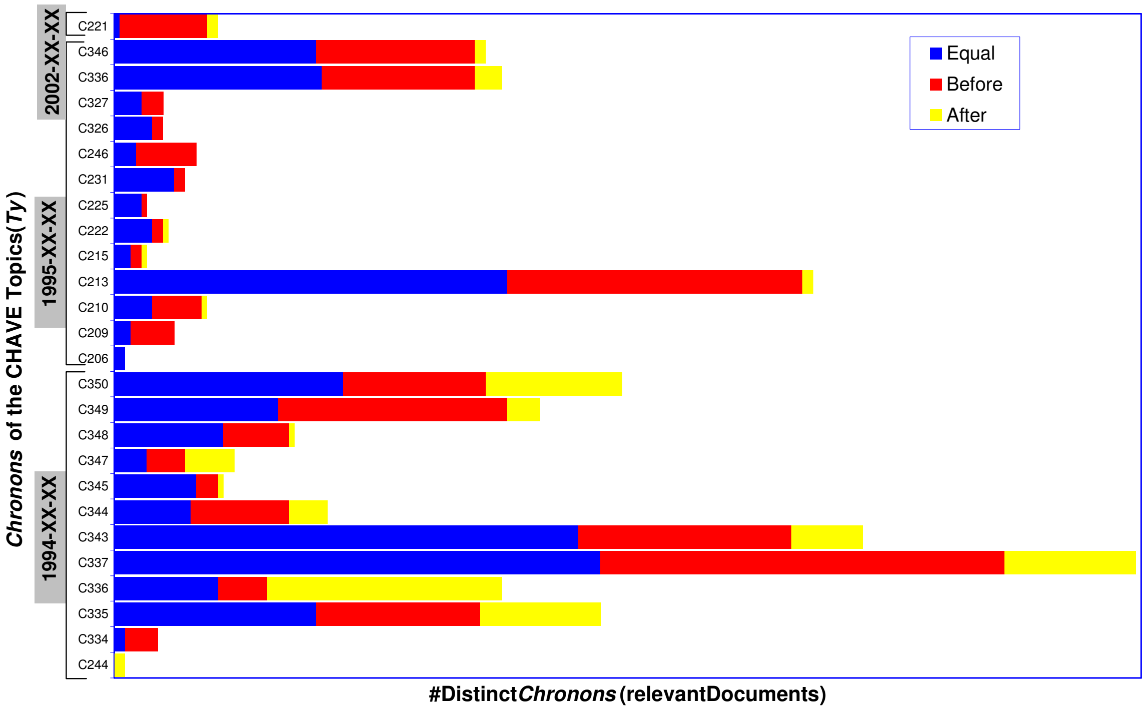


Figure 4.15: *Chronons* of the CHAVE topics represented in the timeline T_y.

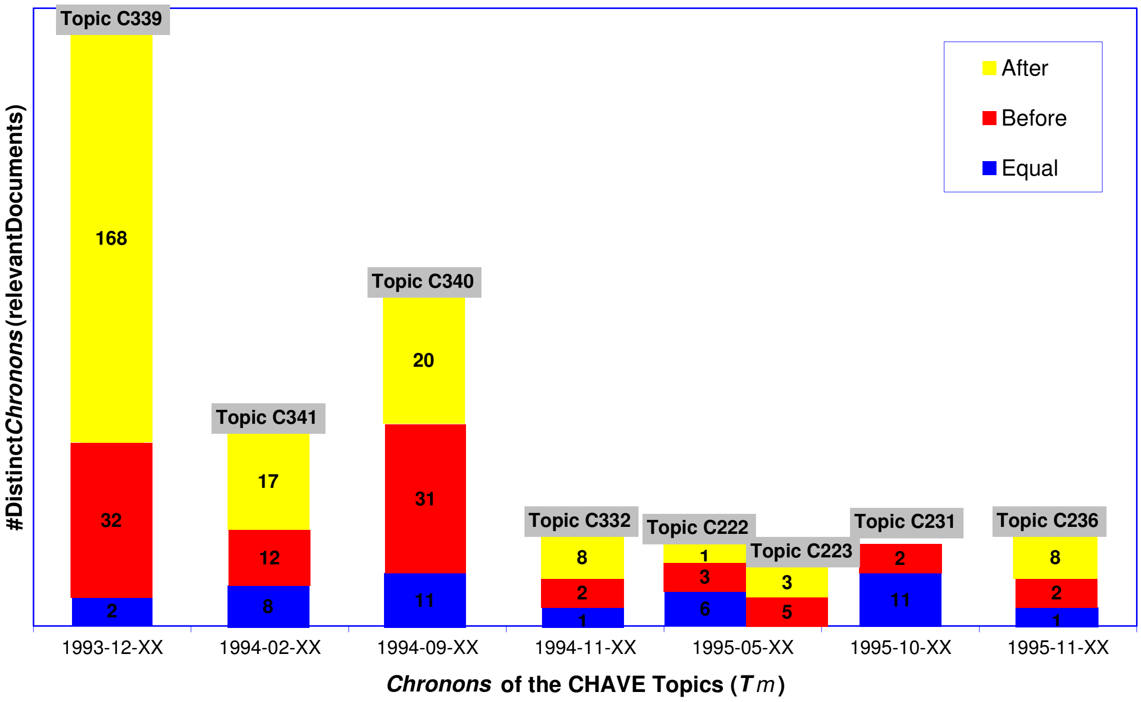


Figure 4.16: *Chronons* of the CHAVE topics represented in the timeline T_m.

Analysis per Topic

In order to know the temporal scope of each topic and to understand how the *chronons* are distributed for the relevant and non-relevant documents, we determined the frequency per *chronon* of relevant and non-relevant documents. For simplicity, the *chronons* that only have a representation in a timeline with the granularity of hours (T_h) were not considered in this analysis.

The time sensitivity of topics was also carried out by the analysis of the temporal information in relevant and non-relevant documents. A topic is time-sensitive when the temporal references, represented as *chronons*, of their documents converges to a date or a period of time. Obviously, the topics with explicit temporal information are already classified as time-sensitive. Table 4.12 displays the number of time-sensitive and time-insensitive topics. Time-sensitive topics are divided into topics with explicit temporal information in the text query and topics with implicit temporal information. We verified that the time-sensitive topics are well represented in two of the topics groups, namely 68% and 50% of topics of the group 1 and group 3, respectively. Group 2 only contains 30% of time-sensitive topics.

	Number of Topics		
	time-sensitive		**time-insensitive**
	***Explicit*TempInf**	***Implicit*TempInf**	
(1) TopicsC201-C250	14	20	16
(2) TopicsC251-C300	0	15	35
(3) TopicsC301-C350	18	7	25

Table 4.12: Time-sensitive classification of the CHAVE topics.

Due to space constraints, it is not possible to discuss every 150 topics. However, as an example, we present the analysis of three topics C222, C254, and C310, representing the three time sensitivity classification of the queries, namely, time-sensitive with explicit temporal information in topic text, time-sensitive without temporal information in topic text, and time-insensitive.

Figure 4.17 and Figure 4.18 show the information of topic C222 with explicit temporal information *1995-05-XX*. We can observe in Figure 4.17 that almost all *chronons* are in the period of time defined by the topic. The *chronons* with the higher frequency of relevant documents are *1995-05-08* and *1995-05-09*. Figure 4.18 only displays the ten *chronons* with higher frequency of non-relevant documents set, since there are much more distinct *chronons* in this document set. We can observe that *1995-XX-XX* is the *chronon* with an occurrence in the most number of non-relevant documents. However, the *chronons* are scattered along the timeline. We verified that

100% of the relevant documents have at least one reference to *May 1995*, while only about 25% of the non-relevant documents have a reference for that date.

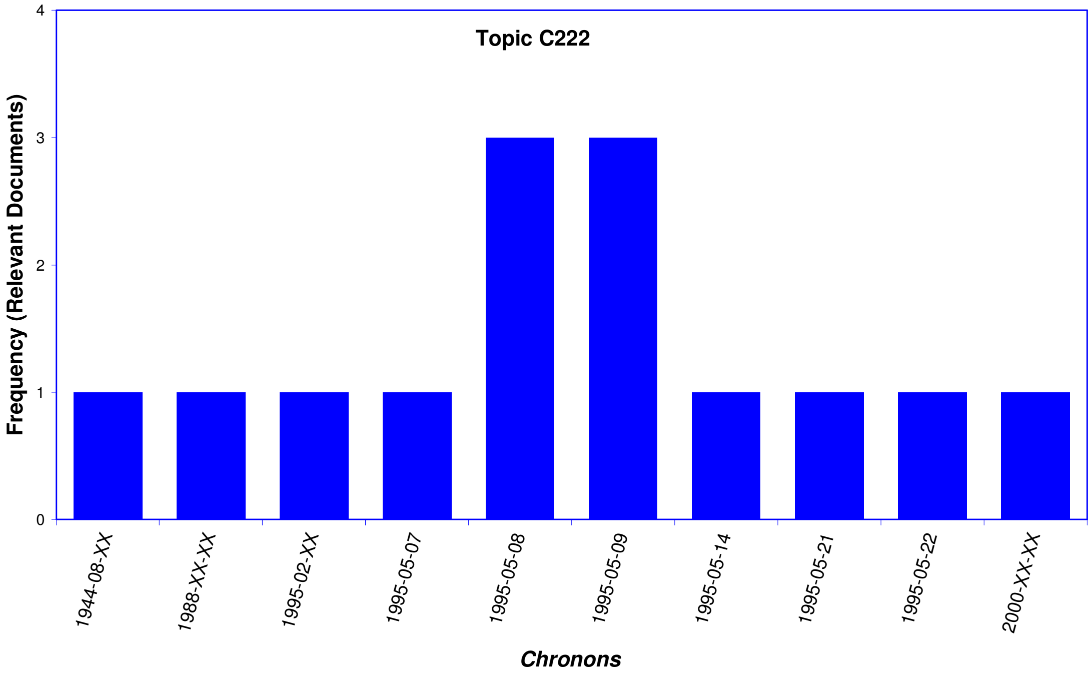


Figure 4.17: Topic C222: frequency of relevant documents per *chronon*.

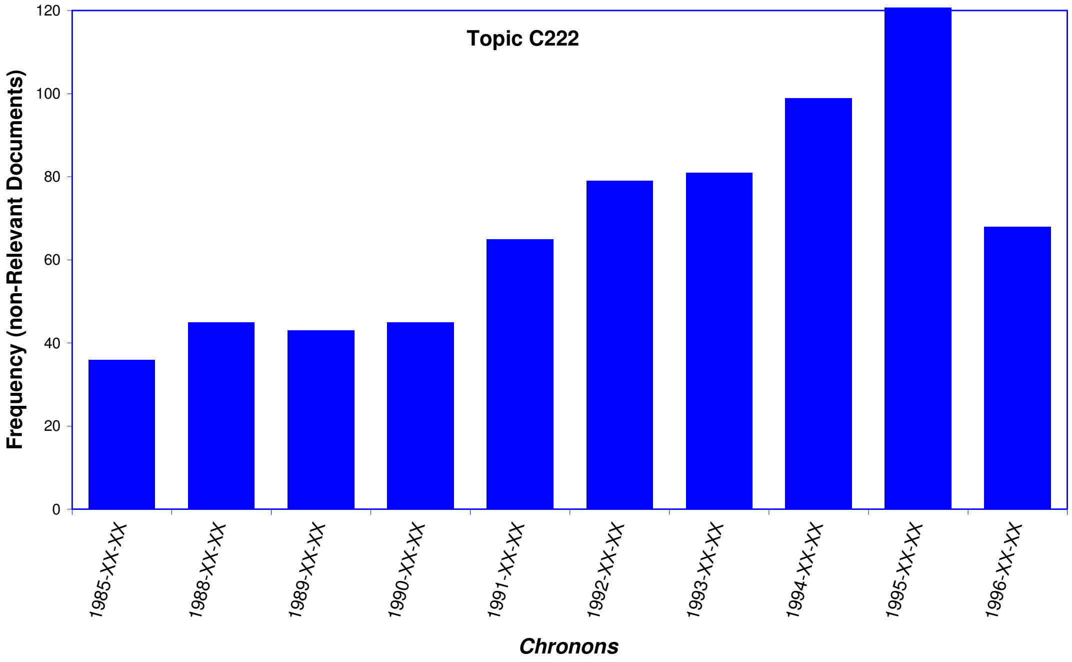


Figure 4.18: Topic C222: frequency of non-relevant documents per *chronon*.

Although topic C254 does not have explicit temporal information, the *chronons* of relevant documents converge to a date. We verified that the temporal references of both relevant documents set and non-relevant documents present the same tendency of the topic with explicit temporal information. In other words, the relevant documents converge to a date unlike non-relevant documents. In topic C254, about 85% of the 129 relevant documents have a reference to *January 1995*, whilst only 25% of the 181 non-relevant documents have a reference to this date and their *chronons* are scattered along the timeline, such as topic C222.

Figure 4.19 shows the number of relevant documents and non-relevant documents of topic C310 per *chronon*. For simplicity, the timeline was limited to the *chronons* which occur in the most documents. As we can see in this chart, the *chronons* in both document sets are distributed along all timeline. Since there is not a concentration of *chronons* around a date, we conclude that this topic is not time-sensitive. The *chronons* which occur in most documents are *1994-XX-XX*, and *1993-XX-XX*. *1994-XX-XX* which occurs in about 20% of the non-relevant documents, and *1993-XX-XX* which occurs in about 10% of the relevant documents.

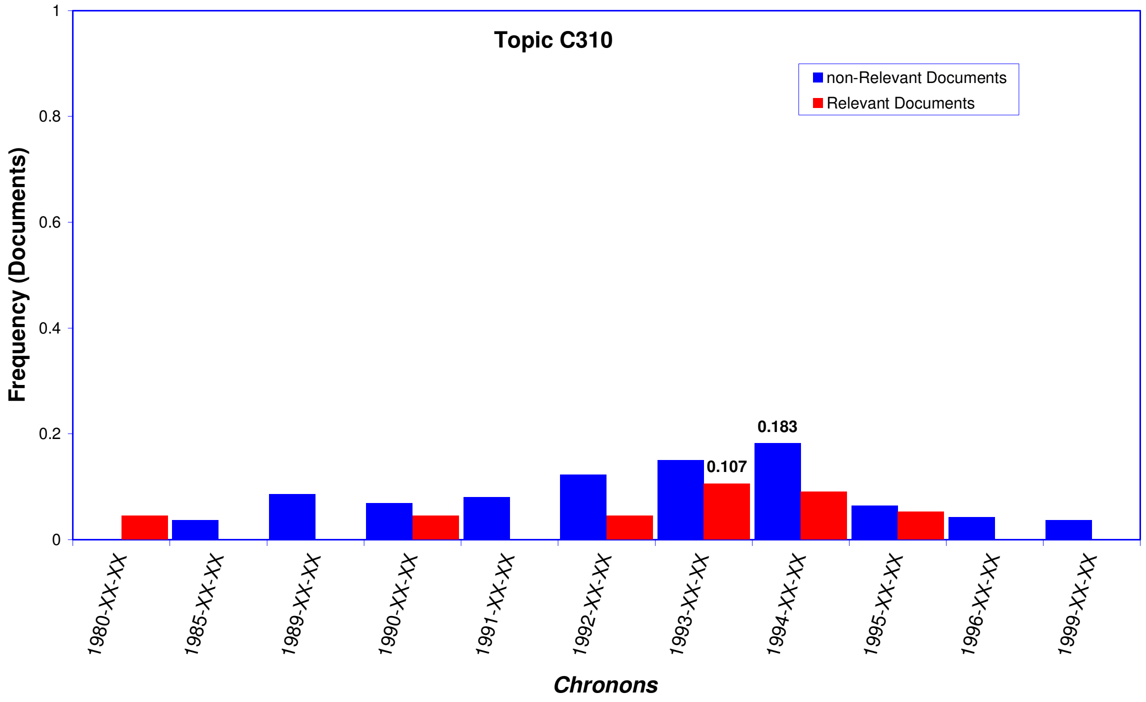


Figure 4.19: Topic C310: frequency of non-relevant and relevant documents per *chronon*.

4.4 Summary

The center of our research is to improve the retrieval effectiveness, taking the advantage of the temporal information found in the content of documents and queries. For this reason, we performed a further analyzed focused on temporal attributes of the collections used along this work: Second HAREM and CHAVE. This chapter presents the main characteristics of the collections, which were created and made available by Linguateca. Some measures were defined to better understand how temporal references are expressed in the text, in order to help to define the models proposed in this thesis. These measures are described and subsequently used in the temporal characterization of the collections.

The results obtained for the Second HAREM Collection show that the temporal information appears in the content of almost all documents (about 98%). We verified that after the resolution of these temporal expressions, this information can be used to split documents into segments, an average of about 7 per document. The results show that we can consider the sentence as the minimum unit of the text segmentation.

By the analysis of the CHAVE collection, we verified that there is normalized temporal information in both topics and documents. About 22% of the topics and 86% of the documents have at least one *chronon*. We observed that in this collection the temporal references of both documents and topics are around the time period of [1994-1995], which is the time period of the publication date of the collection documents. The temporal segments contain, in average, about 69 words and one simple *chronon*.

In order to understand the direct relationship between the words and time, the analysis should be based on words. However, this analysis would lead to an excessively complex processing, since each occurrence of each word would have to be processed. To minimize the processing cost, we analyzed the occurrence of the dates in documents and also in the three different types of topics.

For each one of these types, namely, time-sensitive with explicit temporal information, time-sensitive with implicit temporal information, and time-insensitive, we analyzed the distribution of the dates in relevant and non-relevant documents along timelines. The results obtained shows that the relevant documents of the time-sensitive topics, both with explicit and implicit temporal information, converge to a specific date, as opposed to the non-relevant documents which are dispersed along the timeline.

In addition, the set of obtained results confirms that the CHAVE collection is sufficiently rich to be used in evaluation of temporal information retrieval models.

Chapter 5

Time-aware Information Retrieval

The time-aware information retrieval model provides a re-rank of the retrieved documents set in order to improve the results of queries, both time-sensitive and time-insensitive. This model is based on a temporal segmentation of Portuguese texts and is detailed in this chapter, describing all its characteristics and functionalities. One application of the model is in five new methods of query expansion proposed in this chapter. This chapter also presents a temporal Web crawler, which is another application of the temporal relationship between words used in the proposed model.

5.1 Introduction

The most important research work already carried out so far is essentially based on models where words and time are considered independent. Although, there are also some models that explore the association of events and time, they are based on an assumption that is clearly different from the approach of the model proposed in this thesis. These models only establish relationships between the expressions denoting the events and the time of the event occurrence. In our work, each word of the documents becomes associated to dates.

Our model takes advantage of the relationship between words and time, which is given by the temporal segmentation of texts. The temporality of each word is included in the information retrieval system, adding new indexes. We present some proposals to incorporate the temporal information in the query expansion models.

Our temporal segmentation platform, described in previous chapters, was already applied in query expansion and in temporal Web crawling. However, there are several research opportunities to be explored that can benefit from our work of developing the research platform, such as additional research on temporal focused crawling, document ranking and clustering, just to name a few.

This chapter is organized as follows. Section 5.2 describes our approach of time-aware information retrieval. Section 5.3 explains the application of our approach in order to improve retrieval effectiveness when using query expansion technique. Its evaluation is presented in the following chapter. Section 5.4 describes an adaptation of the time-aware information retrieval approach to a focused Web crawler with time constraints. Section 5.5 finalizes the chapter, emphasizing its main ideas and contributions.

5.2 Time-aware Information Retrieval Approach

Although words and time references (e.g. extracted dates) occur in the same text, to the best of our knowledge, there are no works establishing relationships between them at a text segment level, as the previously presented work in Section 2.3. The advantage of such approaches is the simplicity and the readiness of the existing models of information retrieval.

Our approach is specifically intended to explore these relationships in documents and queries, allowing an explicit association between words and dates, defining the word temporality. This way is possible to know in which date (or dates) occurs a word of a given document.

When we compare our approach with other time-aware information retrieval research works, the first observation is that we establish a temporal relationship between terms. Temporal segmentation of texts gives those relationships through the temporal discontinuities found in documents. Our temporal segmentation algorithm considers temporal information found in the documents content and documents timestamp, in order to determine the document *chronons*. In other models, the *chronons* are also defined as a bag as well as the terms, and there are not any temporal relationships between them.

Most documents report events and related facts that occur during given periods of time. These periods are established by temporal references in the content of documents. Thus, once obtaining all dates from a document, we can associate each date with a set of neighboring words occurring in the document. In this way, the words can be temporally related.

Considering that words with the same temporality can share the description of the facts and events also temporally related, the assumption is that words with the same temporality of the query terms make the correspondent documents more relevant to the query. Thus, the main objective of our approach is to explore the relationship between time and words in order to define a set of methods and heuristics that can improve information retrieval systems. This way, the index data of information retrieval models were enriched with the temporal information of the segments, namely creating a new index structure to store this information, named temporal index, which can be crossed with positional indexes, such as inverted index.

The main drawback is the large amount of index information to support this approach. However, in this phase of the work, we focus our effort only in the effectiveness of the technique, leaving the efficiency problem for a next stage.

In the context of information retrieval and regarding time, queries are classified as time-sensitive and time-insensitive. The first are queries with explicit or implicit temporal references. This means that temporal information is expressed in query text or can be extracted from the query meaning. Recalling the example presented in Section 1.2, the queries *Lisboa 1755*, and *Lisboa terramoto*[27] are time-sensitive, with explicit and implicit temporal information, respectively. These queries focus on the same time period because they address the same event, which occurred on *1st of*

[27] English translation: *Lisbon earthquake*

November, 1755. Time-insensitive queries do not have obvious time-specific information, neither in the user's need, nor in the result set of queries.

Although our approach considers both types of queries, its processing can be different for each of types. The temporal segmentation of documents is used to improve retrieval results, based on ranking formulas considering the time dimension of queries. For time-sensitive queries, the set of documents retrieved are re-ranked by introducing a temporal factor that fits the relevance score of results to the expected reference of time.

Combining the word position in the text and its temporality, we can also define the temporal location of words that can be used to calculate a distance between them. Therefore, temporal information can be used even for time-insensitive queries.

Although temporal information can be incorporated in various contexts, as shown in Section 2.3, we applied our approach to query expansion with a proposal of five methods presented in Section 5.3. In addition, our approach has been used in focused Web crawling, which is fully described in Section 5.4.

5.2.1 Temporal Indexes

Several strategies can be used to keep temporal information in information retrieval index subsystem, which is obtained from the temporal segmentation of texts. A naïve strategy is to enrich the actual document metadata with temporal information, namely start and stop positions of the segments and their associated timestamps. However, this simple approach has the disadvantage of not allowing direct access to documents that match a certain time restriction.

Alternatively, we define one more index, the temporal index, based on temporal segments of documents. For a given query, we can first search the segments matching the query and then combine the score of the segments belonging to the same document.

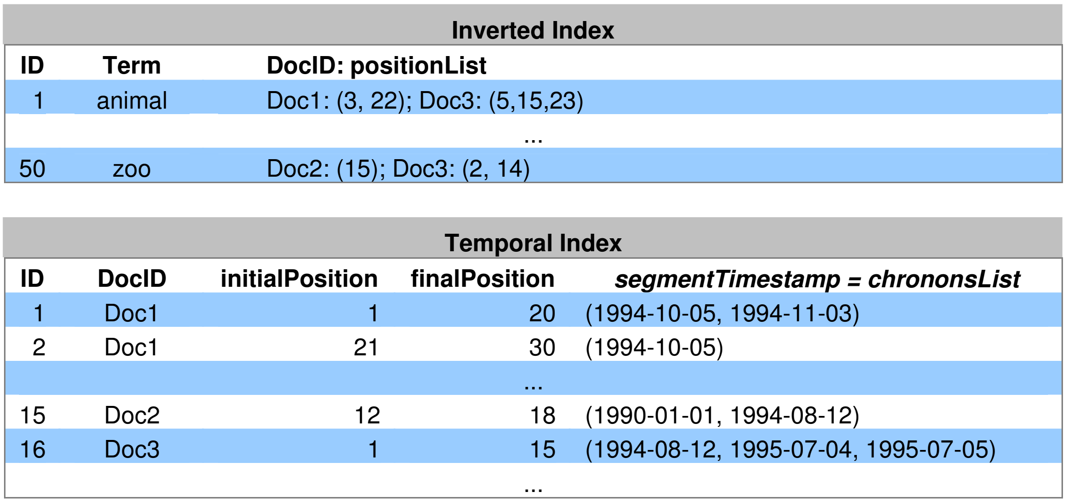
The index construction is a major issue, due to the importance of an efficient organization, since the large amount of index information that must be supported. The metadata for each segment includes a segment identifier, a document identifier, the initial and final position in the document, and the segment timestamp that is a set of associated *chronons*. This allows us to reach the original document and also to have the original relative position of words. The temporality of words is obtained by the conjunction of this index with the traditional inverted index that gives the position of terms within a document.

In this way, considering the positions of words in documents, and the positional limits of segments, it is possible to obtain all the dates associated to each word.

In the indexes construction, index unit is the document temporal segment. From here, we can use vector space model concepts, formulas and weighting methods (see Section 2.2).

Figure 5.1 shows an example of the indexes, in order to help in understanding the approach. For simplicity, the example does not represent the term weights that are also stored in inverted indexes. Therefore, inverted index is composed of term identifier (ID), term, and a list with the position of the term occurrences in each document (DocID: positionList). Temporal index is composed of segment identifier (ID), document identifier (DocID), the initial position and the final position of the segment, and a list of the *chronons* associated to the segment (SegmentTimestamp).

In this example, the term *animal* occurs in positions 3 and 22 of Doc1. The position 3 is in the segment 1 that contains the positions between 1 and 20. The position 22 is in the segment 2 ([21..30]). Thus, this term occurs in the segments 1 and 2, and 16. Its temporality is {1994-10-05, 1994-11-03, 1994-08-12, 1995-07-04, 1995-07-05}. The occurrences of the term *zoo* are in the segments 1, 15, and 16. The temporality of *Zoo* is {1994-10-05, 1994-11-03, 1990-01-01, 1994-08-12, 1995-07-04, 1995-07-05}.



Inverted Index

ID	Term	DocID: positionList
1	animal	Doc1: (3, 22); Doc3: (5,15,23)
		...
50	zoo	Doc2: (15); Doc3: (2, 14)

Temporal Index

ID	DocID	initialPosition	finalPosition	*segmentTimestamp = chrononsList*
1	Doc1	1	20	(1994-10-05, 1994-11-03)
2	Doc1	21	30	(1994-10-05)
			...	
15	Doc2	12	18	(1990-01-01, 1994-08-12)
16	Doc3	1	15	(1994-08-12, 1995-07-04, 1995-07-05)
			...	

Figure 5.1: Indexes example.

5.2.2 Temporal Operators

Based on the concept of the temporal distance between document terms, some temporal proximity operators are introduced. These operators can be used in queries with temporal restrictions. Although based on different information, temporal proximity is analogous to the same concept derived from the relative position of words within a document. So, we can use the same operators with a different semantic. These kinds of operators establish several temporal relationships between document terms, even enabling the use of vague temporal information. Table 5.1 shows some example of temporal operators.

Query Operator	Meaning
keyword(s) [**at**|**before**|**after**] *date*	event occurring *at*, *before* and *after* a given date
keywords(s) **between** *date1* **and** *date2*	event occurring in a time period *between 2 given dates*
keywords1 **/t** *keywords2*	*t* is the maximum temporal distance *between event1 and event2*
keywords1 [**simultaneous**|**before**|**after**] *keywords2*	event1 in *simultaneous, before, after* event2

Table 5.1. Temporal operators.

For the operator ***at***, we need to reach documents in which the keywords occur in a segment tagged with the given date. For ***before*** or ***after*** operators, we need to retrieve documents, including segments whose dates are before or after the given date. For the operator ***between***, we need to find segments with temporal tags between two dates.

For the operator ***/t***, we need to find a document in which the first set of keywords that occurs is within a distance lower than ***t*** relative to the segment of second set of keywords. Some of the operators need precise information but others only need vague temporal information. For instance, the operators ***after***, ***before*** or ***simultaneous*** do not need a date to be formulated.

With this approach, even for queries without any explicit or implicit temporal information, we can use temporal cues for document ranking improvement or penalty. For instance, if the two query terms occur in the same temporal segment of the document, the document ranking must be improved. Otherwise, the document is penalized. Here, the temporal context of document terms is used to score the document.

So far, we have been only talking about very simple strategies of relationship between words and time. The relationships that words can have in temporal domain were not yet explored. There are already a number of research papers that establish temporal correlations between different words or concepts [Amodeo et al., 2011, Radinsky et al., 2011, Whiting et al., 2011]. Most of these studies are concerned with the long-term temporal correlation of words, based on the publication date of the documents. With our temporal segmentation platform, we can extract richer temporal information and therefore establish more realistic correlations. As we are using content dates, the study of short term correlations may also make sense.

The strategy built for this work does not consider the possibility of extracting the temporal information from hypertext content. In this case, the dates of publication of the documents that are referenced may be attached to words that are used in anchor text. This approach allows extracting richer temporal information. Thus, temporal characterization of collections of hypertext as the Web is a major challenge, as a result of this work. The temporal characterization of the Web in turn allows the study of the temporal focused crawling problem.

5.3 Time-aware Query Expansion

The main objective of this work is to improve the results in information retrieval systems, integrating in these systems the relationship between words and time obtained by the content of documents. Considering the query expansion as one of the possible applications, we propose five methods to incorporate the word temporality in the automated query expansion models. The proposed methods are the following: 1) temporal filtering, 2) temporal weighting, 3) temporal aware query reweighting, henceforth temporal reweighting, 4) temporal profile of terms, and 5) temporal profile of pseudo-relevant documents.

All the methods follow the same approach based on the assumption that the terms that occurring in the same temporal segment are stronger for the original query; consequently they deserve greater emphasis. Therefore, these terms assume a more important role in the different steps of the query reformulation.

In order to better understand the details of the methods proposed, the query expansion technique is addressed in the following paragraphs, presenting a brief description and the general algorithm. After that, the details of each method are fully explained. The methods are described according to the various steps identified when we look at the general query expansion algorithm. Note that the proposed methods are not performed in sequence, but they are different alternatives of the query expansion task.

All the methods were implemented, and the experiments carried out and the results obtained are described in Chapter 6. Note that the evaluation of the temporal profile of terms and temporal profile of pseudo-relevant documents methods is not included in Chapter 6. The main reason is that it was not possible to obtain mature evaluation of these methods, given the workload involved and the time available to do it. However, the evaluation of the methods is a part of the on going work, and the results obtained in the first evaluation of the temporal profile of pseudo-relevant documents method are already published [Craveiro et al., 2015].

5.3.1 Query Expansion

The main objective of the query expansion is to increase the quality of the results retrieved from a user's request. In order to try to define the query with more clarity and less ambiguity, usually, the original query is expanded with more terms that have a similar meaning. The new query terms are selected from a pseudo relevance document set, usually the top-ranked documents of the first ranking pass [Rocchio, 1971]. However, the inherent ambiguity of the query and the chance of a polysemy in the query terms, for example, may cause an incorrect expansion. In this way, the additional query terms may do just the reverse of the main objective. These terms can change the focus of the user's intent.

The formulation of effective search requests by users is not an easy task, in part due to some inherent ambiguity of words. The same concept can be expressed using a variety of words and the same word can have more than one meaning. The recall of information retrieval systems can become better by joining synonyms to the query, mainly in thematic collections. Since synonyms can also change the context of the query, in broader collections the recall cannot have the same impact. Many techniques were developed to use thesauri in the query reformulation, either using predefined dictionaries, or automatically creating them. These techniques are included in the global methods, since the automatic query expansion techniques were categorized as global or local.

Global methods are techniques focused on relationships between the words found in a collection, so the new query is expanded or reformulated with semantically similar terms. Local methods expand a query by analyzing the terms included in the top-ranked documents retrieved for the original query [Manning et al., 2008].

Generally, local methods have a better performance than global methods [Carpineto and Romano, 2012]. Indeed, the techniques based on dictionaries, thesauri, or other similar knowledge representation sources, such as WordNet, namely linguistic techniques, are not as effective as the methods based on statistical analysis, due to the inherent ambiguity of the words. As some words have multiple meanings, the word sense disambiguation must be applied to do an exact identification of the word sense, i.e. the meaning of the word. However, local methods can penalize retrieval when just a few relevant documents are in the top-ranked documents retrieved. So, the performance of these methods dependents on the amount of relevant documents that are in the top-ranked documents retrieved [Carpineto and Romano, 2012, Manning et al., 2008, Xu and Croft, 1996].

Local methods rely on relevance feedback which is motivated by the relevance judgments done of the retrieved documents for the original query, as relevant or non-relevant. The common terms from these documents and their weights are used to expand the original query. The most popular algorithm used to compute these weights is Rocchio's method [Rocchio, 1971] that introduces the relevance feedback, given by the user into the vector space model. This algorithm is based on the assumption that the user identifies a set D_r of relevant documents and a set D_{nr} of non-relevant documents. The objective is to approximate the expanded query vector to the centroid of the relevant documents and deviate it from the centroid of the non-relevant documents. Starting from the original query vector $\overrightarrow{q_0}$, the expanded query $\overrightarrow{q_e}$ is given by the following formula:

$$\overrightarrow{q_e} = \alpha \overrightarrow{q_0} + \beta \frac{1}{|D_r|} \sum_{\vec{d_i} \in D_r} \overrightarrow{d_i} - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d_j} \in D_{nr}} \overrightarrow{d_j} \tag{5.1}$$

In Formula 5.1, *α*, *β*, and *γ* are the positive weights associated to each term. The values of *β* and *γ* can be increased according to the confidence on the judged document set; we can consider a higher value for *β* and *γ* when we have a lot of judged documents [Manning et al., 2008].

The automatic query expansion is a particularity of the relevance feedback, also known as pseudo-relevance feedback, blind relevance feedback or retrieval feedback. Unlike the relevance feedback method, this method does not require the classification of the relevant and non-relevant documents given by the user. However, it considers the assumption that the set D_r of relevant documents is given by the top-ranked documents retrieved for the original query. So, in this manner the required user interaction is discarded. For example, in online systems of information retrieval, the information of the previously retrieved documents obtained for a given query can be used by these methods to produce better query statements [Salton et al., 1985].

The process underlying the pseudo-relevance feedback methods is to formulate a new search based on the original query and a set of terms selection from the top-ranked documents retrieved in a first ranking pass. After that, the selected terms are added to the original query with a weight and the expanded query is submitted again to the information retrieval system. An adaptation of the Rocchio's method to pseudo-relevance feedback provides the most popular measure to compute the weight of the terms used for the selection of query expansion terms. Formula 5.2 gives the measure defined by Rocchio (1971).

$$w^*(t \mid q) = \alpha \times w(t \mid q) + \beta \times \sum_{d_i \in D_r} \frac{w(t \mid d_i)}{|D_r|} - \gamma \times \sum_{dj \in D_{nr}} \frac{w(t \mid d_j)}{|D_{nr}|} \tag{5.2}$$

The version of the Rocchio's formula used by pseudo-relevance feedback methods assumes that there is no information about non-relevant documents (γ=0), and considers the assumption that the set D_r of relevant documents is given by the top-ranked documents retrieved for the original query (D_{pr}) as Formula 5.3.

$$w^*(t \mid q) = \alpha \times w(t \mid q) + \beta \times \sum_{d_i \in D_{pr}} \frac{w(t \mid d_i)}{|D_{pr}|} \tag{5.3}$$

In Formula 5.3, *w(t|q)* is the weight of the term in the original query; *w(t|d)* is the weight of the term in the document which is an element of the set of pseudo relevant documents (D_{pr}).

Automatic Query Expansion Process

The process of automatic query expansion starts after the execution of a user's request. The terms used by the user to define his query and the top-ranked documents retrieved for the query are analyzed in order to get the expanded query. This latter query is again submitted to the system to return the results to the user. Note that the first round results are not displayed to the user.

The process of automatic query expansion can be divided into four main steps: data preprocessing, feature generation and ranking, feature selection and query reformulation [Carpineto and Romano, 2012].

In the first step, the data sources used to expand the original query are formatted. This step is required when the data sources used for expanding the query are not the same source where the query was searched. When the data source is the same data collection, each document is already represented as a set of weighted terms, by using the inverted index file. Usually, the inverted index also stores the term positions to provide proximity-based search. The top-ranked documents retrieved in the result list of the original query are the basis information for most of the query expansion techniques. However, the data sources can also be, for example, a dictionary, the Wikipedia, anchor texts, etc.

The second step provides the candidate expansion features and a ranking based on the scores assigned to these candidates. Following an approach inspired by the Rocchio's method for relevance feedback [Rocchio, 1971], all the terms of the top-ranked documents retrieved in the first pass are collected. After the removal of stopwords and the stemming, a score is assigned to each word by a defined term weighting formula. Several formulae have been proposed to provide the importance of a term in the query, which is given by the assigned term weight. The different methods proposed for the ranking of terms follow two main approaches.

One approach uses the weight of the term computed for document ranking in the first pass. Usually, Formula 5.4 is used to compute the score assigned to each term, which is given by the sum of the weight of the term ($w(t \mid d)$) in each document of the pseudo relevant document set (D_{pr}).

$$\sum_{d_i \in D_{pr}} w(t, d_i) \tag{5.4}$$

The main problem of this approach is that the weight of each term can better express its importance in the whole collection than in the original query given by the user.

A different approach, which follows a probabilistic model, is focused on the analysis of the difference between the distribution of terms in the set of pseudo-relevant documents and the distribution of the same terms in the whole collection [Carpineto and Romano, 2012]. Several functions following the latter approach have been used by query expansion models, such as binary independence retrieval model [Robertson and Jones, 1976], Robertson selection value [Robertson, 1990] and Chi-square [Doszkocs, 1978]. Carpineto et al. (2001) proposed an approach that relies on relative entropy by using the Kullback-Leibler distance between the two distributions. Wong et al. (2008) obtained very good results in a recent experimental study with term ranking functions based on chi-square statistics and relative entropy (Kullback-Leibler distance).

According to Carpineto and Romano (2012), the performance of the whole system does not show significant differences when the score for the term ranking is computed by different functions, unless these functions are also used to reweight the expansion terms.

The third step carries out the selection of terms to be added to the original query. The modified query with the original terms and the chosen terms is used to be submitted again to the information retrieval system. Traditionally, the first *n* terms of the ranking, given by the last step, are the chosen terms. There is not a general consensus among researchers about the optimal number of terms to be included, but typically the *n* value is defined between 10 and 30. The other used approach is the definition of a threshold for the score assigned to the terms, instead of the number of terms. The last approach is mainly used when the ranking function follows a probabilistic model [Carpineto and Romano, 2012].

In the last step of the automatic query expansion, the original query is reformulated, adding the new terms chosen in the previous step. Usually, it is assigned a new weight to each term of the expanded query, although some approaches do not perform the query reweighting as explained below. Thus, the approaches can be classified into two categories: approaches with and without query reweighting.

The simplest approach does not perform any query reweighting. The query is expanded with the original terms and the new terms. After that, this expanded query is submitted to the system without computing any weight to each of its terms. This means that all the terms have the weight set to 1. So, considering the Rocchio's formula (see Formula 5.3), α and β are set to 1.

The approach followed by most systems computes the weight of each term of the expanded query using the Rocchio's formula, duly adapted for pseudo-relevance feedback (see Formula 5.3). This formula can be employed both in the choice of the terms to be used in the expansion and in query reweighting. But as these two tasks are independent, there are also systems which employ a different formula for the term ranking, such as Robertson selection value, and then apply Formula 5.3 to all terms of the expanded query.

Another approach employs functions based on probabilistic models to compute the weight of the term (see explanation of the second step). The weight assigned to the terms for the selection of the terms to be included in the expanded query is also used in the query reweighting, instead of Rocchio's weights formula. In this case, Formula 5.3 is updated to:

$$w^*(t \mid q) = \alpha \times w(t \mid q) + \beta \times score(t) \tag{5.5}$$

In Formula 5.5, the *score(t)* is the weight assigned to term *t* given by the term-weighting formula also used in step two.

Automatic Query Expansion Algorithm

Figure 5.2 shows a general algorithm, considering a broad view of the query expansion process. The two important parameters of the algorithm are: *k*, and *n*. *k* is the number of relevant documents for the original query *q0* defined by the user. This parameter is used in the definition of the expanded query *qE*. The maximum number of terms in the expanded query is defined by the parameter *n*.

Roughly speaking, the algorithm can be described as follows: first, it is necessary to obtain a set of terms to be included in the expanded query *qE*; and then, the original query *q0* is reformulated considering these terms and their scores.

In order to obtain the potential terms, the result list of the initial query *q0* is processed to get a document set *D* composed of the top-*k* most relevant documents to the original query *q0* (step 3). All the terms of these documents are extracted, but they are exposed to a refining, such as stopwords removal, stemming or other filtering type.

```
(1) Specify parameters: $k$, $n$
(2) Submit the query $q0$ with the set of terms ($T_0$) $tq0_1$, …, $tq0_m$ defined by
     the user
(3) Get a document set $D$ composed of the top-$k$ most relevant documents to
     the query $q0$
(4) Let $T_1$ be a set of the terms extracted from the document set $D$
(5) For each term $t$ of the terms set $T_1$, compute its score ($score(t)$)
(6) Rank the terms set $T_1$, in decreasing order
(7) Pick-up the top-$n$ terms $t_1$, …, $t_n$ of the ranked list of terms ($T_1$)
(8) Compute the weight for each term $tq0_1$, …, $tq0_m$ (original terms) and $t_1$,
     …, $t_n$ (new terms)
(9) Reformulate the original query $q0$ in an expanded query $qE$ with the
     terms $tqE_1$, …, $tqE_{m+n}$ ($TqE= T_0 \cup T_1$) and their weights
(10) Submit the new query $qE$
```

Figure 5.2: Query expansion algorithm.

For example, Amati (2003) only puts a term in the set of potential terms T_1, if the term exists in the content of at least two documents (step 4). The choice of the n terms to be included in the expanded query qE is based on a score assigned to each term of the set T_1. The top-n terms with high score are chosen to be included in the new query (steps 5, 6 and 7). There is a panoply of formulae to compute the score of the terms. For example, the most popular formula which is Rocchio's formula (see Formula 5.3) or the probabilistic formulae proposed by Amati (2003).

The new query qE is composed of the terms used in the original query $tq0_1$, …, $tq0_m$ and the other terms chosen in the last step t_1, …, t_n ($TqE= T_0 \cup T_1$) (steps 8-9). Some approaches define different weights for these terms by using formulae to compute them, such as Rocchio's weights formula (see Formula 5.3). Finally, the reformulated query qE is submitted to the system (step 10).

5.3.2 Temporal Filtering

The temporal filtering method is focused on step 4 of the query expansion algorithm presented in Figure 5.2. The other steps are performed following the original algorithm. Therefore, the step 4 is defined by:

```
(4.1) Let $T_1$ be a set of the terms extracted from $D$
(4.2) Remove all terms without a temporal relationship with query
       terms $tq0_1$, …, $tq0_m$ from $T_1$
```

Initially, the set of candidate terms T_1 is composed of all the existent terms in the content of the documents belonging to the document set *D*. The main objective of the temporal filtering method is to decrease the number of candidate terms in order to only keep the most important terms that describe their documents.

Usually, a great number of systems apply stopwords removal and stemming to the set of terms T_1. After that, our method applies another filtering process to this set. In this filtering, only the terms which co-occur with the original query terms $tq0_1$, …, $tq0_m$ in the same temporal segments are kept in the set of terms T_1; all other terms are removed from this set. In this way, only the new terms with a temporal relationship with the query terms are considered potential terms of the expanded query *qE*.

Figure 5.3 presents an example of the temporal filtering of terms. Before the temporal filtering, the set of candidate terms T_1 is composed of *term3*, *term4*, and *term5* which are the terms extracted from pseudo-relevant documents. In the temporal filtering processing, *term5* is excluded from T_1, since its temporality does not match the temporality of the original query terms. As result, the set of candidate terms is given by *term4*, and *term5*.

```
Set of original query terms:                              T0 = {term1, term2}
Set of terms extracted from top-k most relevant documents to q0
(candidate terms):                                      T1 = {term3, term4, term5}

Temporality of the query term term1:      temp_term1 = {1994-10-22,1994-10-23}
Temporality of the query term term2:      temp_term2 = {1994-10-22,1995-05-01}
Temporality of the original query terms:
                                    temp_T0 = {1994-10-22,1994-10-23,1995-05-01}

Temporality of the candidate term term3: temp_term3 =  {1994-10-22,1994-10-23}
Temporality of the candidate term term4: temp_term4 =  {1995-05-01,1995-05-02}
Temporality of the candidate term term5: temp_term5 =  {1995-12-15}

After temporal filtering, set of candidate terms:         T1 = {term3, term4}
```

Figure 5.3: Example of temporal filtering.

5.3.3 Temporal Weighting

The temporal weighting method introduces a new formula to be used in the computation of terms score, referred in Figure 5.2 as step 5. The other steps are performed following the original algorithm. In fact, Step 5 is divided into two sequence steps: first, compute the term score, and then, modify this score using a time weighting.

```
(5.1) For each term $t$ of the terms set $T_1$, compute its score
      $score(t)$
(5.2) Compute for terms in $T_1$ a $score^*(t)$ based on $score(t)$
```

1) First, the computation of the term score *score*(*t*) is carried out using a conventional formula, such as Formula 5.3 or other formula, for instance, a formula available in the Divergence from Randomness framework [Amati, 2003].

2) Then, in order to promote the terms which have a temporal relationship with the terms of the original query $tq0_1$, …, $tq0_m$, the score previously computed is modified. The score of a term *t* changes according the temporal distance to the query terms $tq0_1$, …, $tq0_m$. In fact, the score is only modified if the term *t* and the terms of the original query $tq0_1$, …, $tq0_m$ are not in the same temporal segment. The score becomes inversely proportional to this temporal distance *td*(*t*), as shown by the following formula:

$$score^*(t) = score(t)\times\frac{1}{1+td(t)} \qquad\qquad td(t)\in\{0,1\} \tag{5.6}$$

The temporal distance *td*(*t*) is used to penalize the term score if the term *t* does not belong to the same temporal segments of the query terms. *td*(*t*) is *1* for the maximum temporal distance between the term *t* and the query terms which can be simply when the term *t* and the query terms are not in the same temporal segment. Following a discrete approach, *td*(*t*) is *1* when *t* and the query terms are not in the same segments; otherwise 0.

Indeed, this method does not exclude terms, but they will be in different position of the ranking (step 6). The terms with a temporal relationship with the query terms are promoted in the ranking, obtaining more probability to be picked-up in step 7.

5.3.4 Temporal aware Query Reweighting

The method presented in this section is focused on the query reformulation, represented by steps 8 and 9 in Figure 5.2, without interfering with the terms selection, unlike the others. Indeed, the other steps are performed following the original algorithm.

The original query *q0* is rewritten with the terms of the set *TqE* ($TqE = T_0 \cup T_1$). This means that the reformulated query is composed of the original terms $tq0_1$, …, $tq0_m$ and the terms defined to be included in the expanded query t_1, …, t_n. Each of one can have a weight in the expanded query.

Our proposal is to give different weights to the terms that co-occur in the same temporal segments of the original query terms, namely increasing them with a weight *δ*.

Considering the Rocchio's formula (see Formula 5.5) to associate weights to the terms in the reformulated query, Formula 5.7 computes the weight to apply to each term, considering *α* and *β* positive weights associated to the initial and the new query terms, respectively.

$$w^*(t \mid q) = \alpha \times w(t \mid q) + (\beta + \delta) \times score(t) \tag{5.7}$$

The parameter *δ* is only assigned if the term *t* co-occurs in the temporal segments of the original query terms and its value must be limited as follow: $\beta + \delta \leq 1$. Generally, the initial query terms obtained the maximum weight (*α=1*). *β* and *δ* are tuning parameters. The weight of a new term is 1 (*β*+*δ=1*) when it obtains the same importance to the query as the initial query terms. The weight of the term *t, score(t)*, is given by the term-weighting formula considered in step 5 in Figure 5.2.

5.3.5 Temporal Profile of Terms

The method presented in this section is focused on the selection of the terms (steps 4, 5 and 6 of Figure 5.2) using their temporal profiles. A temporal profile is built with the temporal information gathered from the top-*k* most relevant documents to the original query *q0* (document set *D*), namely content time and publication time. Indeed, according to our temporal approach, this temporal information is represented by the timestamp of both documents and text segments of where the term occurs.

The temporal profile of a term is the probability distribution over time, considering the temporal scope of the document set *D*. The timeline is defined with the most frequent granularity of *chronons* of this document set. The terms ranking only include terms that occurred in the areas of the timeline with the highest concentration of terms. In other words, it include the terms positioned in peaks, assuming that a peak around a specific date make it an important point in time for the query topic. The terms ranking is based on their probability of occurrence at those points in time. The chosen ones are *n* terms that have a higher probability of occurrence. If there are no peaks, then the query is not sensitive to the time for a specific document collection. Jones and Diaz (2007) classified this type of queries as *atemporal queries*. So, this method is not applied for this particular case.

5.3.6 Temporal Profile of Pseudo-Relevant Documents

Diaz and Jones (2004) proposed an approach to improve the average precision of queries. The retrieved documents for a given query are promoted only if their publication dates are positioned in the time period relevant to that query. This period is determined by the analysis of the temporal profile of the query, which is created from the distribution of the publication date of the retrieved documents.

Based on this approach, we define the temporal profile of a document that is the relevant time period for the document, given by the content time and the publication time of this document. This information about the document is useful to select the top-*k* most relevant documents to the original query *q0* (document set *D*). Therefore, this method applies to the step 3 of Figure 5.2 which can be defined by:

```
(3.1) Compute peak dates from the ordered retrieved documents (set
      R) to q0
(3.2) Remove from set R documents without occurrences in peak dates
(3.3) Get the document set D composed of the top-k most relevant
      documents to q0  (set R)
```

As the method explained in Section 5.3.5 above, only the documents with a representation in the peak dates are considered for the selection of the top-*k* most relevant documents. The peak dates are computed as the outliers in the distribution of the number of documents. Documents without an occurrence in the peak are removed from the set of the pseudo-relevant documents to the original query *q0* (set *R*). If there is more than one peak, then an intersection of the documents in all the highest peaks is applied. If there are no outliers, this set *R* remains unchanged.

The date of publication and the *chronons* extracted from the documents content of the set *R* are considered for computing of peak dates. First, the number of documents for each date is determined. Subsequently, the peaks are selected based on an outlier detection criterion, as the one defined by Chauvenet (1960).

In order to improve our methods, namely, temporal filtering, temporal weighting, and temporal aware query reweighting, this method can be applied as a complementary strategy to be used with them.

5.4 Temporal Focused Web Crawling

In the last years, the exploration of temporal dimension has been object of a deep research in various areas, such as retrieval models, clustering, etc. However, the use of time constraints has not due received enough attention in the focused Web crawling area.

Web crawling is used to update the Web content or indexes of Web Sites, mainly search engines, so that the request of users can be processed more quickly. The search engines index the pages downloaded by the crawlers. Web crawler starts with a list of URLs to visit. For each URL visited, the crawler identifies all hyperlinks in the Web page and adds them to the list of URLs to visit. The URLs are recursively visited according to a set of policies. As it is not possible to download the entire Web, the crawler must prioritize the Web page to download. Therefore, more efficient search mechanisms are needed. Not all information on the Web is required, only the correct part of it. The Web crawling with constraints is one of the ways to solve the scalability problem of the crawling.

Recently, Pereira (2013) developed a Web crawler that can work with some time constraints, taking advantage of the temporal references in the content of documents. This work was based on our assumption that words with the same temporality can make the documents more relevant for the query, with suitable adaptation to Web crawling context. So, the temporal Web crawler was created following the assumption that links in the temporal segments with the same temporality are more important than the other links.

The main objective of this crawler is to cross the Web searching information in a temporal scope previously defined, ensuring that the Web pages downloaded by the crawler are within that temporal scope. Figure 5.4 shows the architecture of the temporal Web crawler. In general, the architecture of this crawler is similar to the architecture of a traditional Web crawler. The modules of the architecture of a traditional Web crawler are fully explained by Pant and others (2004). The only difference between the two architectures is the temporal analysis required to execute in each page after the temporal segmentation of this page. This difference is represented in Figure 5.4 by a blue rectangle with a discontinuity line, which surrounds the two new modules.

The module *segmentation of the Web page* represents the temporal segmentation of the Web page that is being processing. This module is carried out using our tools described in the previous chapters. The result of the processing is the Web page divided by temporal segments and represented in a XML document.

The temporal analysis is performed with the temporal segments of the Web page. This module is responsible for the identification of the URLs that are in temporal segments with the same temporal scope defined to the crawler. When a temporal segment with that time constraints is found, then the

URLs that are inside this segment are marked as valid URL. The valid URLs are processed by the next module, *URL prioritizing*, otherwise the URLs are discarded.

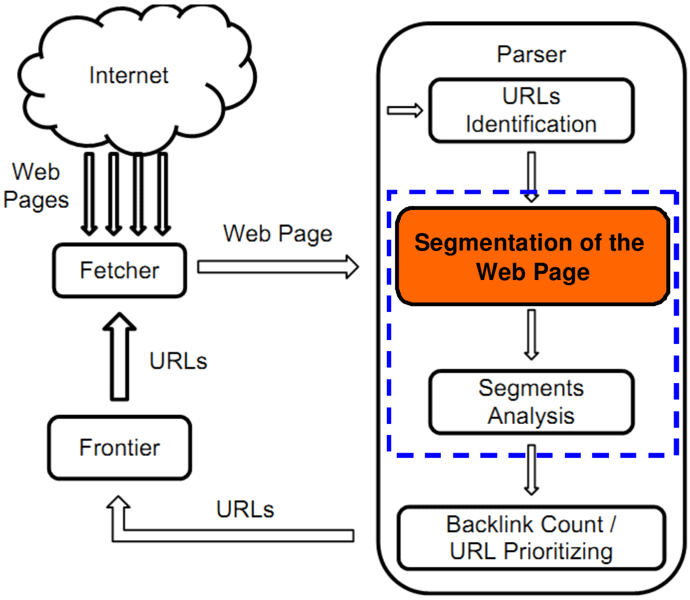


Figure 5.4: Temporal Web crawler architecture, adapted from [Pereira, 2013].

The temporal Web crawler was implemented using *Crawler4j*[28], which is a generic crawler developed in Java language. It provides a very simple way to use and customize, since all the crawling and downloading processes can be inherited from the original classes. For that reason, all the logic working of the crawler is not changed as well as the way of the interaction between the modules.

To evaluate the temporal crawler, Pereira (2013) used the Portuguese Wikipedia, which is hosted under the domain *http://pt.wikipedia.org/*. Only the pages inside the same domain were considered. He defined two groups of seeds, based on the important marks of the Contemporary History with different temporal scopes: World War II (WWII), from 1939 to 1945, and the attacks of 9/11 (2001-09-11). The crawler was configured to download 5000 pages, and starts with 10 seeds for each of group.

First, the Web crawling was carried out with a generic crawler without any time constraints, and then, the time constraints were configured with 1939-1945 (WWII) and 2001 (9/11), for the

[28] https://code.google.com/p/crawler4j/ [June 30th, 2013]

temporal Web crawler. The results of the generic crawler were used as baseline for the temporal crawler.

In this evaluation, Pereira (2013) only considered the metrics of precision and recall. A Web page is relevant to be downloaded by the temporal crawler if it is referenced by links positioned in segments whose timestamps are in the interval of the time constraints. To calculate the total number of the relevant pages in collection, he considered the union of relevant sets from the temporal crawler and the generic crawler for the same collection.

The results obtained in this evaluation were published by Pereira (2013) and Pereira et al. (2014). The author observed that the temporal crawler achieved better results than the generic crawler in both metrics and for each group – WWII and 9/11. The recall obtained when 5000 files are downloaded by the generic crawler was about 40% in both groups, while the temporal crawler obtained a recall about 70% and 60%, in WWII and 9/11, respectively.

The value of precision has an even greater difference, with more emphasis on the WWII group. Approximately 25% was the precision achieved by the generic crawler, while the temporal crawler obtained about 80% at 100 files and about 45% at 5000 files. The precision obtained by the generic crawler on the 9/11 group is about 60% at 100 files and about 25% at 5000 files. The values obtained by the temporal crawler are about 70% and 60%, respectively.

Although the temporal crawler needs some improvements [Pereira, 2013], we can already conclude that with these auspicious results the temporal crawler is a promising concept to explore and represents one interesting application of our research work. Obviously, there is a need for a more comprehensive research and evaluation of the temporal crawler approach.

We can also conclude that the application of our approach to include time constraints in the focused Web crawling context, following the assumption that the information (words or, in this case, links) expressed in the same segments tend to be more important for the results, is a great choice.

5.5 Summary

Temporal segmentation of texts has the particularity to establish relationships between words and time and thus, between the words in time domain. This relationship is established from the publication date and the temporal information extracted from the content of documents. Documents are divided into temporal segments based on temporal discontinuities found in texts.

This chapter presents a new proposal that takes advantage from this relationship between words to improve retrieval effectiveness. The presented approach proposes the use of temporal information on two specific tasks. One is the automatic expansion of queries and the other is the focused Web crawling.

In the case of query expansion, the traditional queries expansion algorithm is presented first. Then, five different query expansion methods are introduced, identifying the steps on which the base algorithm is modified. The modifications are based on the increase of the importance of the terms in expanded query temporally related with the terms of the original query.

In focused Web crawling, the downloading of Web pages is based on time constraints, considering not only the modified and/or creation dates, but also the temporal segments of the pages. It is intended to download only pages that match the established timing constraints.

Chapter 6

Temporal Query Expansion Methods Evaluation

This chapter details the evaluation carried out to validate our proposed methods for improving the query expansion. The methods use previously extracted temporal information, as explained in the last chapter. For each method, a set of experiments have been performed in Terrier platform with time-sensitive and time-insensitive queries of the CHAVE collection. A brief description of Terrier, and the changes carried out are the first issues presented in this chapter. Then, each of the experiments is fully described, as well as the considered settings. The results obtained are fully discussed.

6.1 Introduction

In order to evaluate the performance of the proposed methods for improving query expansion, described in Section 5.3, we developed all the required software using Terrier platform[29], namely Terrier 3.5 [Macdonald et al., 2012]. Terrier is an open source modular platform for the rapid development of large-scale information retrieval applications. The abridged description of this platform, developed at the University of Glasgow, Scotland, is presented in the next section.

Concerning the developed tools to extract temporal information, described in the early chapters, as we have stated our main focus was not efficiency. Furthermore, the development carried out in the Terrier was not optimized, although there was a concern regarding the simplicity of algorithms and the choice of hash tables as the data structures used. Therefore, in this stage, the evaluation of efficiency was not considered, and was only focused on the retrieval effectiveness.

Note that although all the proposed methods was implemented, the evaluation of the temporal profile of terms and temporal profile of pseudo-relevant documents methods are not described in this section. Given the workload involved and the time available to do it, it was not possible to obtain mature evaluation of the temporal profile of terms and temporal profile of pseudo-relevant documents methods. However, the evaluation of the methods is a part of the on going work, and the results obtained in the first evaluation of the temporal profile of pseudo-relevant documents method are already published [Craveiro et al., 2015].

For each of the other temporal query expansion method, namely, temporal filtering, temporal weighting, and temporal aware query reweighing, we carried out a set of experiments with the CHAVE collection, described in Section 4.3. This is the only available collection of texts in Portuguese, with the three components required for the evaluation of the effectiveness of information retrieval systems: documents, topics and relevance judgments. The temporal characterization of the CHAVE collection reported in Section 4.3 and published by Craveiro et al. (2015) confirms the usability of the collection for temporal *ad hoc* information research.

Although there are three groups of topics available, we only chose two of them – C251-C300 and C301-C350. The main reason for our choice was that the relevance judgments of the third group do not take into consideration all the documents of the CHAVE collection, but only the set of PUBLICO, representing about 50% of the collection documents. Obviously, with a different set of documents, it is not possible to compare the results.

[29] http://ir.dcs.gla.ac.uk/terrier/ [September 20th, 2015]

Typically, in order to evaluate the retrieval effectiveness when using query expansion, each query is executed with and without query expansion, and then the two retrieved result sets are compared. In such case, the result set without query expansion is the baseline. Since our base model is Bose-Einstein 1 (Bo1) (the default query expansion model of Terrier), we also considered a run with this query expansion model to be compared with the results obtained by our proposed methods, which is a stronger baseline. The configurations of this run and the baseline are described in Section 6.4.

The two most common measures used to evaluate the retrieval effectiveness are the *average precision* for each query, also known as topic, and the *Mean Average Precision* (MAP) for one group of queries [Carpineto and Romano, 2012].

Average precision is the sum of the precision at each relevant document of the ranked retrieved result set, divided by the number of relevant documents. Formula 6.1 is used to compute this measure, where *n* is the number of retrieved documents, and *i* is the position of the document in the ranked retrieved result set. *P(i)* is the precision at position *i* in the retrieved set. *rel(i)* is a binary function with a value of 1 if the document at position *i* is relevant, and otherwise 0 [Manning et al., 2008].

$$Average\,\mathrm{Pr}\,ecision = \frac{\sum_{i=1}^{n}(P(i)\times rel(i))}{\#\,\mathrm{Re}\,levantDocuments} \tag{6.1}$$

Mean Average Precision is computed by the sum of the *average precision* obtained for each query, divided by the number of topics submitted to the system (Formula 6.2) [Manning et al., 2008].

$$MAP = \frac{\sum_{i=1}^{m} Average\,\mathrm{Pr}\,ecision(i)}{m} \tag{6.2}$$

In Formula 6.2, *m* is the number of topics submitted to the system, and *AveragePrecision(i)* is the average precision for the topic *i*.

In the context of the actual information retrieval systems, the result of queries can be a set of thousand relevant documents, and the users will be not interested in get all of them. So, recall is not so important, but Precision at *k* documents (Precision@*k*) is still useful. Precision@*K* is the precision at *k* retrieved documents, which gives the percentage of relevant documents in the top-*k* retrieved documents.

According to Carpineto and Romano (2012), to evaluate the robustness of the system is as important as to compute the retrieval performance, because the performance of the retrieval when using query expansion can be very different from query to query. Even when most of queries are improved, some others are penalized.

The evaluation of robustness is a common practice, by using the standard measure *Robustness Index* (RI), which is given by Formula 6.3:

$$RI = \frac{IMP - PQ}{nQ} \tag{6.3}$$

In Formula 6.3, *IMP* is the *Improvement metric* (IMP) [Xu et al., 2009], given by the number of individual queries that were improved by using the query expansion method. *PQ* is the number of penalized queries, and *nQ* is the total number of queries.

6.2 Terrier Platform

Terrier (TERabyte RetrIEveR)[30] [Macdonald et al., 2012, Ounis et al., 2005, Ounis et al., 2006, Ounis et al., 2007] is the platform chosen to do the experimental tests of the query expansion methods proposed in the last chapter. Terrier is a software package, developed in Java, for the rapid development of Web, intranet and desktop search engines. This developing platform provides the ability to index and query document collections in various formats, as well as the evaluation of standard TREC collections.

Terrier was considered one of the five search engines with best performance, achieving good results in the experiments performed by Middleton and Baeza-Yates (2007).These authors studied the performance of different open source search engines, which was focused on the following parameters: precision and recall, indexing time, index size, resource consumption, and searching time.

Zettair was the search engine with the better performance. However, it is well known that there has been much information retrieval development since 2007, which may have changed the rankings for the search engines obtained in the study performed by Middleton and Baeza-Yates (2007). Since we observed at Zettair site[31] that there is not new information since 2009, we considered that it would not be a good choice.

Besides the good results obtained by Terrier, we verified that this platform had continuous upgrades. Terrier is also very well structured, as well as it offers very good documentation[32], making it easy to add new retrieval models. As Terrier also supports the Portuguese language, we chose it for our work.

[30] http://ir.dcs.gla.ac.uk/terrier/ [September 20th, 2015]

[31] http://www.seg.rmit.edu.au/zettair/ [September 20th, 2015]

[32] http://terrier.org/docs/v3.5/ [September 20th, 2015]

In the following paragraphs, we present a brief description of Terrier.

Before indexing, Terrier identifies the terms, which were represented as UTF characters, to be included in indexes from a stream of text. This task includes the tokenization, according to the language being dealt with. The appropriate *tokeniser* for Portuguese is UTFTokeniser, which uses a Java Character class to determine what characters are valid in the indexing of terms.

Terrier also provided the configuration of stopwords. The terms configured as stopwords are not included in the indexes. The Terrier parsing includes an optional configuration for stemming. All stemmers from the Snowball stemmer project[33] were incorporated in Terrier. *PortugueseSnowballStemmer*, a stemmer for Portuguese, is also available.

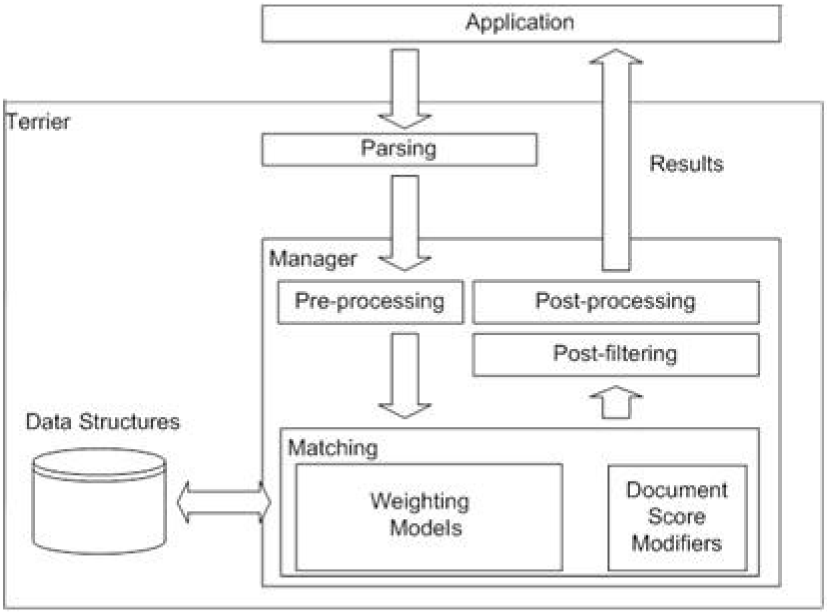


Figure 6.1: Retrieval architecture of Terrier [Ounis et al., 2006].

Figure 6.1 shows the retrieval architecture of Terrier [Ounis et al., 2006] with the representation of the interaction of components in the retrieval phase. Data Structures represent all the indexes of the system. The manager component receives the query previously parsed, and performed the preprocessing, for instance, removing stopwords, applying stemming, etc. The matching component initializes the Weighting Model configured and is responsible for computing the

[33] http://snowball.tartarus.org/ [September 20th, 2015]

document scores, considering the query submitted. Before the result set of the relevant documents will be sent to the application, they can still be processed to be changed.

Query expansion uses the top-*n* of the retrieved documents set, for further processing. So, first, it is required to obtain such document set. In Terrier, the set of retrieved documents can be further modified by applying post-processing or post-filtering.

Post-processing is the appropriate option to implement query expansion, since it allows changing the original query and run it again.

Post-filtering is the last step before the results become available to the application, where the information request was submitted. A post-filter is used to remove documents from the retrieved result set that do not satisfy a given condition. For example, to limit the number of retrieved documents from the same Web site, in order to give more diversity to the results [Ounis et al., 2006].

Note that the indexation must be carried out before the beginning of the retrieval process. Terrier provides five main structures: lexicon, inverted index, document index, direct index, and meta index:

- Lexicon stores the terms of document collection and the corresponding document and term frequencies.
- Inverted index stores the posting lists of each term, which means the identifiers of the documents and their corresponding term frequencies, and the position of terms within a document (optional).
- Document index stores information about each document collection, namely, the identification, the length in number of tokens, and a pointer to the corresponding entry in the direct index.
- Direct index stores, for each document, information about its terms, namely, the term identifier and the corresponding frequency.
- Meta index stores additional information (metadata) about each document which must be configured, such as the docno or the URL.

A variety of retrieval models are already available in Terrier, such as the classical TF-IDF, the Okapi BM25 probabilistic weighting model, Ponte-Croft's language modeling and a vast number of DFR models [Ounis et al., 2006]. However, as the purpose of our experimental tests is query expansion, and it is orthogonal with the retrieval model, we can choose any one.

Terrier also included some query expansion models, such as the Rocchio's method [Rocchio, 1971], and the models proposed by Amati and van Rijsbergen (2002) and Amati (2003), namely Kullback-Leibler divergence, Chi-square divergence, Bose-Einstein 1 (Bo1), and Bose-Einstein 2 (Bo2). These term weighting models, which follow the DFR models, are used to identify the terms to be added to the query. Note that they provide a parameter-free approach, being an alternative to the Rocchio's formula (see Formula 5.3). Nevertheless, new models can also to be created in Terrier.

The default query expansion model of Terrier is Bo1. This term weighting model is one of the best-performing implemented by Terrier, namely improving MAP [Amati, 2003]. McCreadie et al. (2009) also testified its suitability for application using various methods, in particular, to generate good feedback documents. For this reason, we chose this model as our base model for the weighting of terms.

In the following paragraphs, we present a brief overview of Bo1. A full description of this and the other models were published by Amati and van Rijsbergen (2002) and Amati (2003).

Bo1 assigns a score to give the importance of each candidate expansion term. This score is calculated by the divergence between its information content and the probability of its frequency in the set of pseudo-relevant document following a Bose-Einstein distribution, given by Formula 6.4 [Amati, 2003]:

$$w(t) = tf_x \times \log_2 \frac{1+f}{f} + \log_2(1+f) \tag{6.4}$$

In Formula 6.4, tf_x is the frequency of the term t in the set of pseudo-relevant documents. The term f is the average frequency of the term t per document obtained in the whole collection, which is calculated by $\frac{TF}{N}$. TF is the frequency of the term t in the collection, and N is the number of documents in the collection.

Before the reformulated query to be submitted again to the retrieval process, it must be reweighting. Terrier provides two options: one by defining a value for the Rocchio's parameter β [Rocchio, 1971] (see Formula 5.3), and the other by using a parameter-free approach [Amati, 2003].

Terrier also requires a direct index before starting a post-process for query expansion. The main reason is that in general, the query expansion models need the frequency of terms computed for each document given by this index, in order to compute the weights of the candidate terms to be added to the original query.

Basically, Terrier provides three easy and simple ways to change or extend the query expansion. First, the formula used to weight the candidate expansion terms can be changed, implementing a subclass of *QueryExpansionModel*. Second, the gathering process of candidate terms is carried out from all documents of the pseudo-relevant document set, as one large "bag of words". This process can be changed by extending the *ExpansionTerms* class. Finally, Terrier uses the top-*k* retrieved documents to select the pseudo-relevant documents, but this selection can also be modified implementing a subclass of *FeedbackSelector*.

Query expansion in Terrier also enables the configuration of other aspects. Table 6.1 shows the configurable properties provided by Terrier to use a query expansion model.

Configurable Properties	Description
trec.qe.model	The name of the query expansion model to be used.
parameter.free.expansion	The parameter-free option is or is not activated.
rocchio.beta	The value of Rocchio's parameter, when the option of parameter-free is *false*.
qe.feedback.selector	Class(es) that select(s) the pseudo-relevant documents from the top-ranked retrieved documents.
qe.expansion.terms.class	Class(es) that select(s) terms to be added to the original query. These terms were obtained from the pseudo-relevant document set.
expansion.documents	The number of top-ranked documents to be considered pseudo-relevant documents by the query expansion model.
expansion.mindocuments	A candidate term must exist in a minimum number of documents before it can be weighed, and ranked, to be used in the process of terms selecting.
expansion.terms	The number of the highest weighted terms to be added to the original query.

Table 6.1. Terrier configurable properties for query expansion[34].

Terrier also includes a module to evaluate the retrieval performance of the tested system. This module is a Java implementation of the main functionalities of the *trec-eval* tool for TREC *ad hoc* tasks. So, Terrier provides a system evaluation by computing the following measures: MAP, Precision@ rank *N*, interpolated Precision and R-Precision [Ounis et al., 2006]. Besides that, it makes available the average precision for each query.

[34] http://terrier.org/docs/v3.5/ [September 20th, 2015]

6.3 Terrier Platform Extensions

In this section, we present a brief description of the changes made in Terrier to implement and evaluate our methods to improve query expansion by using temporal information, henceforth tmpQE.

Indeed, since Terrier has a good structure of classes, the development was carried out by creating a new Java classes. Basically, each new main class was created to implement one tmpQE method. The main classes were created in the package *org.terrier.querying*, although it was required to create other classes, namely in the packages *org.terrier.indexing* and *org.terrier.structures*.

In addition to the direct index, tmpQE methods require another index to store the temporal segment information of each document. The temporal index was implemented by the class *temporalIndex* in the package *org.terrier.indexing*.

The temporal index stores for each segment, the identifier of document, the *chronons* list and their boundaries, which are represented by the positions of the first term, and the last term of the segment. Since the inverted index gives the original position of terms within a document, the temporal index working in conjunction with that index provides the association between words and dates.

Figure 6.2 shows the class diagram of the extensions made for query expansion in this work. All the new classes are created in the package *org.terrier.querying*. For simplicity, the class diagram only displays the new main classes, which are illustrated by the boxes in green. Each class is displayed with the principal data structures used by methods, and the override methods, which have an underlined name.

For a better understanding, the other Terrier classes, which have a direct relationship with the new classes, are also represented in the diagram by the boxes in grey. Abstract classes are displayed in italic.

The new classes are subclasses from the two main classes made available in Terrier. Subclasses of the *FeedbackSelector* only select the pseudo-relevant documents to be considered in the query expansion process. The classes that must work with the terms to be used in the reformulated query are subclasses of the *ExpansionTerms.*

All the data structures were created in the package *org.terrier.structures*. These structures were implemented by using hash tables, as the other structures already implemented by Terrier. In fact, hash tables are considered as one of the most efficient data structures for searching, since the hash key can give direct access to the data.

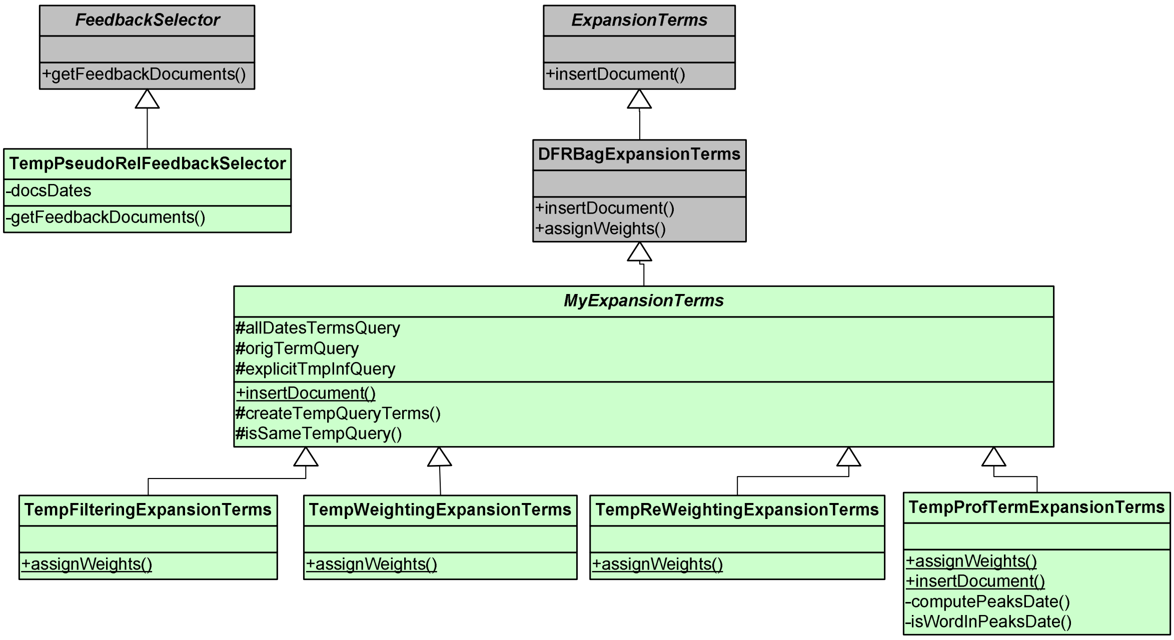


Figure 6.2: Class diagram of the query expansion extensions in Terrier.

The *insertDocument* method adds the information about the pseudo-relevant documents in structures used by Terrier query expansion. This information is obtained from the indexes given by a document identifier. The temporal information is processed by the *createTempQueryTerms* method.

The *createTempQueryTerms* method stores in the *origTermsQuery* structure all the *chronons* found in the pseudo-relevant documents associated to each query term. So, each term query is the key of a list of *chronons*. The *allDatesTermQuery* structure is a set composed of all distinct *chronons*, which were stored in the *origTermsQuery* structure. For queries with the explicit temporal information, this method also build-up the *ExplicitTmplnfQuery* with the *chronons* found in the query text.

Considering that the temporal space is a set of discrete points on the time axis, representing all the *chronons* associated to the word, the matching between the temporal space of the query and the temporal space of a given term is implemented by the *isSameTempQuery* method.

TempFilteringExpansionTerms, *TempWeightingExpansionTerms*, *TempReweightingExpansionTerms*, and *TempProfTermExpansionTerms* are classes that implement the tmpQE methods temporal filtering, temporal weighting, temporal aware query reweighting, and temporal profile of terms, respectively (see Sections 5.3.2, 5.3.3, 5.3.4, and 5.3.5). The *assignWeights* of each of these classes is the Java method that implements the corresponding tmpQE method.

The *TempProfTermExpansionTerms* class has two more methods, *computePeaksDate* and *isWordinPeaksDate*, than the other subclasses of *MyExpansionTerms*. The first method (*computePeaksDate*) distributes the occurrence of terms along the timeline, represented by distinct *chronons* of the pseudo-relevant documents. After that, it determines the *chronons* with the most terms occurrence. The other method (*isWordinPeaksDate*) verifies if a given term occurs in a peak date.

The *TempPseudoRelFeedbackSelector* class implements the tmpQE method described in Section 5.3.6. The main structure used by this class is *docsDates*. This structure is built-up with all the distinct *chronons* that are associated to the set of top documents retrieved for a given query. So, *docsDates* stores, for each *chronon*, a list with the identifier of the documents that contains the *chronon*. The method *getFeedbackDocuments* requires the information given by *docsDates* to obtain the pseudo-relevant documents.

6.4 Evaluation

With the temporal segmentation of documents, each segment obtains one or more *chronons*, whenever possible. Each word of the segment is associated with that *chronons*. Therefore, when a word occurs in a document, it has a given temporal space, which is a set of the *chronons* associated to that word.

Two words co-occur temporally if there is a non-empty set in the intersection of their temporal spaces. The hypothesis stated in this work is that this co-occurrence represents an important link between the words that can be exploited in various contexts and applications. Our proposed query expansion methods, described in Section 5.3, are based on this assumption. Such methods exploit temporal relationships between the words.

In fewer words, the first method (temporal filtering) eliminates all candidate terms that do not co-occur temporally with the original query terms. The second method (temporal weighting) penalizes the score of the terms outside the temporal space of the original query terms, pushing them to the end of the ranking. The third and the last evaluated method (temporal aware query reweighting) does not interfere with the selection process of the terms, unlike the first two. Thus, the set of terms for the expanded query are the same. However, the weight of a term that is in the same temporal space that the original query terms is increased.

The other two proposed methods, namely, temporal profile of terms and temporal profile of pseudo-relevant documents, are not referred in this section. Although we have performed some experiments with these methods, they are not enough to be included in this evaluation. Preliminary results of the temporal profile of pseudo-relevant documents are already published [Craveiro et al., 2015].

To validate our hypothesis assumed to improve retrieval effectiveness by using query expansion, temporal filtering, temporal weighting, and temporal aware query reweighting were evaluated by comparing their performance with the retrieval model TF-IDF, and with the query expansion model Bo1.

Two sets of *chronons* are created during the processing of a given query, in order to establish the temporal relationship between the terms of the original query and the candidate terms to include in the expanded query. They are *queryTerms_chronons*, the set created for the query terms, and *candidateTerm_chronons*, which is created for each term of the pseudo-relevant documents set.

The *chronons* are obtained from the segments of the pseudo-relevant documents. The timestamps of the segments where the term occurs are represented in *candidateTerm_chronons*. The *queryTerms_chronons* set is composed of all *chronons* that were associated to the query terms.

For example, for the query C350, the *candidateTerm_chronons* set of the term *tricampeã* (three-time champion), and the *queryTerms_chronons* are:

```
candidateTerm_chronons(tricampeã)={1995-05-01,1994-XX-XX}
queryTerms_chronons={1994-10-23,1995-05-01,1994-10-22,1994-05-04,
                                  1994-05-05,1994-05-03,1982-XX-XX,1994-XX-XX}
```

So, the candidate term *tricampeã* shares its two *chronons* with the query terms.

In the design of the experiments, we took into account the time sensitivity of the queries to verify the importance of the words temporality considered in our methods. So, for each of the tmpQE methods, we performed the following experiments with different sets of topics, taking into account the implicit or explicit temporal information of the collection topics:

- Experiment A. All topics of the collection (time-sensitive and time-insensitive);
- Experiment B. Only time-sensitive topics;
- Experiment C. Only time-sensitive topics with explicit temporal information in the text of the queries;
- Experiment D. Only time-insensitive topics.

Temporal classification of the test collection topics was done manually by two people. Anyway, the automatic temporal classification of queries is an interesting and important aspect that should be pursued in the future. Indeed, the temporal characterization of the CHAVE collection reported in Section 4.3 and published by Craveiro et al. (2015) marks the beginning of the query classification study. Indeed, in this phase of the work the temporal classification of topics was only important to the design of the experiments, since we want to evaluate the same processing both in time-sensitive and time-insensitive queries.

In fact, we have already taken a step forward in temporal classification of queries, by carrying out the temporal characterization of CHAVE that includes a time sensitivity study of the topics. This study focused on the analysis of the distribution of dates in relevant and non-relevant documents along timelines, for each query types, namely, time-sensitive with explicit temporal information, time-sensitive with implicit temporal information, and time-insensitive (see Section 4.3.1). It was important for understanding the time dimension in documents.

Topics with explicit temporal information can be automatically detected. By explicit temporal information, we mean temporal expressions expressed in the topic text that are mapped into *chronons*. Table 6.2 presents one topic of each set in order to better understand the concept of time sensitivity, and the difference between explicit and implicit temporal information.

Time-sensitive topics

Topic C259
Title: Urso de Ouro (*Golden bear*)
Description: Encontrar documentos mencionando filmes galardoados com o Urso de Ouro no Festival de Cinema de Berlim. (*find documents that mention Golden Bear award-winning films at international festival of Berlin*)

Time-sensitive topics with explicit temporal information

Topic C326
Title: Prémios Emmy Internacional *(International Emmy Awards)*
Description: Encontrar informação sobre os vencedores do Prémio Emmy **de 1995** em programação televisiva internacional. (*find documents about the Emmy award winners* ***in 1995*** *in international television*)

Time-insensitive topics

Topic C255
Title: Viciados na Internet (*Internet addict*)
Description: O uso frequente da Internet produz habituação? (*can the frequent use of Internet produce dependence?*)

Table 6.2. Example of topics classification according to their time sensitivity.

C259 is a time-sensitive topic, since its content is about an event that occurs every year in specified dates. This means that there is temporal information associated to this topic although it is not expressed in the text topic. This is implicit temporal information. The topic C255 is not time-sensitive. The subject of this topic cannot be associated to defined time periods. In the topic C326, the time period is explicitly defined (in 1995) for the information need. This means that this topic has explicit temporal information.

In this section, we describe the experimental setting for each experiment and the results obtained. Preliminary results obtained in the performed experiments are already published [Craveiro et al., 2014a, Craveiro et al., 2014b].

6.4.1 Settings

Collection

As mentioned earlier, all the experimental tests were carried out using the CHAVE collection with the 100 topics considering their title and their description. These topics were grouped into four different sets: *TopicsSet1*, *TopicsSet2*, *TopicsSet3*, and *TopicsSet4*.

- *TopicsSet1* is composed of all 100 topics, from C251 to C350;
- *TopicsSet2* is composed of all the time-sensitive topics, which are the following 40: C257, C259, C262, C265, C266, C267, C277, C279, C280, C282, C284, C287, C290, C292, C296, C305, C308, C313, C316, C326, C327, C332, C333, C334, C335, C336, C337, C338, C339, C340, C341, C342, C343, C344, C345, C346, C347, C348, C349, and C350;
- *TopicsSet3* is a subset of the *TopicsSet2* with explicit temporal information in the text of the queries. These topics are the following 18: C326, C327, C332, C334, C335, C336, C337, C339, C340, C341, C343, C344, C345, C346, C347, C348, C349, and C350;
- *TopicsSet4* is composed of all the time-insensitive topics, which are the other 60 topics of the collection.

Metrics

The retrieval effectiveness of each method was measured by MAP for the top 1000 retrieved documents, Precision@10, Precision@15, and Precision@20. The Average Precision for each topic was also considered. RI and IMP were used for the robustness evaluation. Section 6.1 gives the explanation of the metrics.

Terrier Parameters

Table 6.3 shows the defined properties in Terrier configuration for the experimental tests performed with the tmpQE methods.

Terrier Properties	Value
block.indexing	True
tokeniser	UTFTokeniser
termpipelines	Stopwords,PortugueseSnowballStemmer
TrecQueryTags.process	TOP,NUM,PT-TITLE,PT-DESC
trec.model	TF_IDF
matching.retrieved_set_size	1000
trec.qe.model	Bo1
parameter.free.expansion	True or false
rocchio.beta	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}
qe.feedback.selector	PseudoRelevanceFeedbackSelector
qe.expansion.terms.class	{TempFilteringExpansionTerms, TempWeightingExpansionTerms, TemReWeightingExpansionTerms, TemProfTermExpansionTerms}
expansion.mindocuments	2
expansion.documents	3
expansion.terms	{10, 20}

Table 6.3. Terrier properties configured for the evaluation.

In order to obtain the temporal space of a given word, the information stored in temporal index must be used together with the position of terms within documents, which must be stored in inverted index. As the storage of this information is optional in inverted index, the property *block.indexing* was set to *True*.

The appropriate *tokeniser* for Portuguese is UTFTokeniser. Terrier was configured to apply *PortugueseSnowballStemmer*, the Porter stemmer for the Portuguese language in stemming and to ignore stopwords during indexing and query parsing.

The experimental tests were carried out using TF-IDF as retrieval model, but it can be any model. Note that, the model is not important for our analysis, as referenced by Amodeo et al. (2011) in their work also performed with query expansion models. The retrieved document set for each topic was limited to the top 1000 documents.

As previously mentioned, Bo1 is the base-model used in our experimental tests, which means that the property *trec.qe.model* was defined to Bo1.

In the experiments with the temporal aware query reweighting method, *parameter.free.expansion* was set to False, since the main objective of this method is to increase the weight of terms in the same temporal space of the original query terms. So, *rocchio.beta* was varied between 0.1 and 0.9 (see Table 6.3), considering the requirement that $\beta + \delta \leq 1$. Its increment was limited to 0.1, due to the huge number of possible experiments, considering the parameters that can be changed. As *rocchio.beta* is only configured for the evaluation of temporal aware query reweighting method, its variation is explained in the corresponding experiments section.

For the other methods, the property *rocchio.beta* was set to False, since the *parameter.free.expansion* was set to True. This configuration was defined to take the advantage of parameter-free query expansion formula [Amati, 2003].

The property *qe.expansion.terms.class* must be configured with the java class name, which implements the tmpQE methods used in the experiment.

We define that a term to be considered in the expansion must appear in at least two of the retrieved documents, since Ounis et al. (2006) considered this as the optimal value in *ad hoc* retrieval. So, the property *expansion.mindocuments* was set to 2.

The number of the pseudo-relevant documents considered for the query expansion (*expansion.documents*) was defined as 3 (*Terrier default value*).

The maximum number of terms to be included in the reformulated query (*expansion.terms*) was set to 10 or 20, depending on the performed experiment. Note that, although there is not an optimal value for this parameter, typically this value is defined between 10 and 30 [Carpineto and Romano, 2012].

In experiments A, with temporal filtering and temporal weighting methods, we performed two experiments per each of these methods, configuring *expansion.terms* with 10 and 20. We verified that the results of these two methods followed the same pattern than the model Bo1 when the configuration changed from 10 to 20 terms. The latter are somewhat better in the three methods. Therefore, for the other experiments, we set to 10 terms in order to work in the more restrictive conditions.

Baselines

For each set of topics, we created a baseline with the results obtained by retrieval a maximum of 1000 documents using the TF-IDF model, without query expansion. The baselines are *noQE100*, *noQE40*, *noQE18*, and *noQE60*, with the *TopicsSet1*, *TopicsSet2*, *TopicsSet3*, and *TopicsSet4*, respectively.

We also considered the results obtained by a run per set of topics configured with the query expansion model Bo1, the base model. Its configuration is displayed in Table 6.4. These runs are *Bo1QE100*, *Bo1QE40*, *Bo1QE18*, and *Bo1QE60*, according to the set of the topics used.

Query Expansion Properties	Value
trec.qe.model	Bo1
parameter.free.expansion	True
qe.feedback.selector	PseudoRelevanceFeedbackSelector
qe.expansion.terms.class	DFRBagExpansionTerms
expansion.mindocuments	2
expansion.documents	3
expansion.terms	{10, 20}

Table 6.4. Query expansion properties for the *Bo1QE* runs.

6.4.2 Experiments A (100 topics)

In the following, we present the details of each experiment carried out with all topics of the CHAVE collection (*TopicsSet1*). Each experiment includes the results obtained in each of these experiments, and their comparison with the reference values obtained with the system configuration described in the previous subsection. The best results obtained by each tmpQE method are compared at the end of this subsection.

The objective of these experiments is to verify how important the temporal relationship between words used by our tmpQE method is, when applied in time-sensitive and time-insensitive queries.

Note that the retrieval system for the 100 topics of *TopicsSet1* should find 5,581 relevant documents of the 100,000 retrieved documents.

Experiment A.1 – Temporal Filtering

In this experiment, we evaluated the performance of our temporal filtering method, which removes all terms without a temporal relationship with the terms of the original query. In order to verify how strong this relationship is, we study the number of *chronons* associated to a term that matches the *queryTerms_chronons* set by the variation of parameter *j* between 1 and 10. The higher this value, more restrictive becomes the temporal condition, and therefore the stronger the temporal relationship between a term and the query terms.

Besides that, we also considered two possible values for the number of terms (*n*) to be included in the reformulated query, which were 10 and 20.

Table 6.5 shows the number of relevant retrieved documents, MAP, Precision@10, Precision@15, and Precision@20 obtained from retrieval with and without query expansion, namely *noQE100*, *Bo1QE100*, and the temporal filtering method (TmpF), respectively. The best values of each column appear in bold.

	#Relevant				
	retrieved	**MAP**	**P@10**	**P@15**	**P@20**
noQE100	4450	0.322	0.484	0.438	0.407
Query Expansion, ***n=10***					
Bo1QE100	4824	0.361	0.498	0.471	0.449
Temporal Filtering					
(*) j=1	**4826**	0.361	**0.502**	0.471	0.451
(*) j=2	4823	**0.362**	0.501	0.477	0.452
j=3	4814	0.362	0.497	**0.483**	**0.454**
j=4	4779	0.358	0.494	0.473	0.449
j=5	4740	0.351	0.491	0.465	0.437
j=6	4703	0.350	0.491	0.466	0.434
j=7	4693	0.343	0.487	0.456	0.428
j=8	4685	0.342	0.485	0.459	0.425
j=9	4596	0.336	0.485	0.455	0.417
j=10	4574	0.335	0.489	0.452	0.420
Query Expansion, ***n=20***					
Bo1QE100	4853	0.365	0.501	0.476	0.452
Temporal Filtering					
(*) j=1	4849	0.366	**0.505**	0.479	0.456
(*) j=2	**4856**	**0.368**	0.503	0.484	0.455
j=3	4835	0.364	0.498	**0.485**	**0.460**
j=4	4796	0.360	0.496	0.472	0.449
j=5	4753	0.352	0.489	0.473	0.439
j=6	4714	0.350	0.489	0.468	0.436
j=7	4703	0.344	0.485	0.457	0.430
j=8	4689	0.342	0.486	0.459	0.427
j=9	4595	0.336	0.486	0.455	0.418
j=10	4574	0.335	0.489	0.452	0.421

Table 6.5. Results of temporal filtering with 100 topics.

We observed that the two query expansion methods *Bo1QE100*, and temporal filtering improved the effectiveness. Comparing the results obtained with *n=10* and *n=20*, both methods achieved better results when the reformulated query had the greater number of terms (*n=20*), although with residual gains (around 0.5%-1.5%). We verified the same behavior in these two methods when the number of terms changes from 10 to 20. However, TmpF achieved somewhat higher gains than *Bo1QE100*. The difference is residual, around 1%.

All the results obtained by the temporal filtering method are better than *noQE100*. The Precision@10 obtained the lowest gain (approximately 4%), while the improvement of others is approximately 9%-14.5%. MAP obtained the highest improvement, 14.3% in the configuration *n=20*, and 12.6% in the other configuration.

In Table 6.5, the results obtained the TmpF method when *j≤2* are marked with *(*)*. All these results are greater than the results of *Bo1QE100*, with gains around 0.5%-2.6%. The values of precision achieved the best gains, while MAP achieved only residual differences around 0.5-1%.

The best results of the temporal filtering method were obtained when *j<4*, as shown by Figure 6.3. The MAP significantly decreases when *j≥4*, as well as Precision@10, Precision@15, and Precision@20. As expected, a very limited number of terms was obtained with the increasing of *j*. We should take into account that terms were obtained only from 3 documents, and most documents do not have more than 4 *chronons*, as presented in Section 4.3.1.

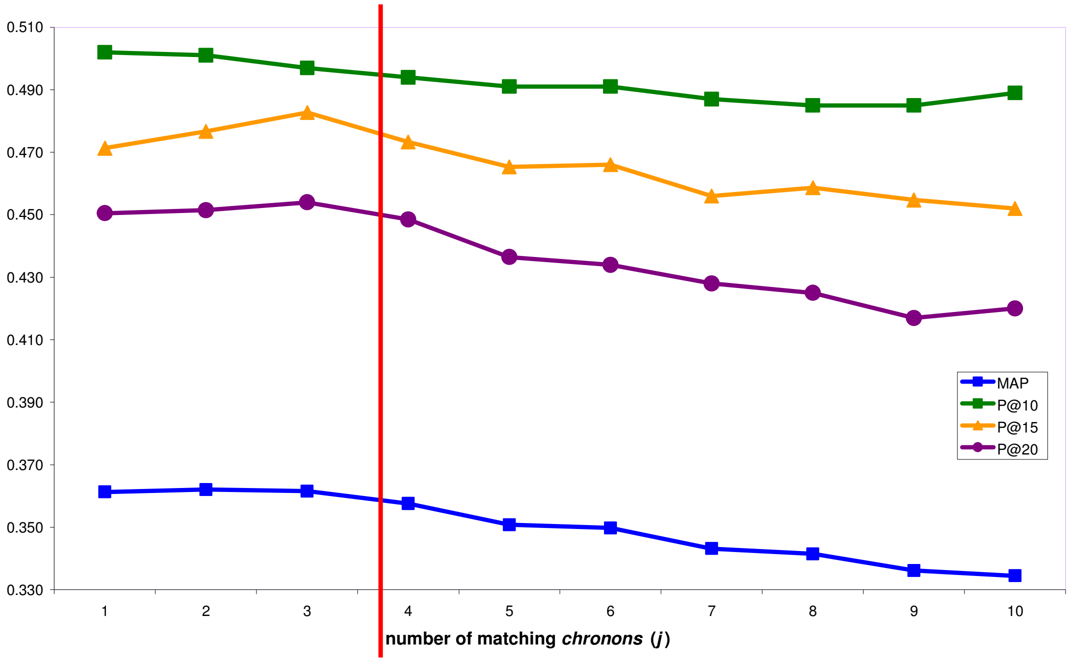


Figure 6.3: Temporal filtering with *n=10*: MAP, Precision@10, Precision@15, and Precision@20.

Experiment A.2 – Temporal Weighting

In the evaluation of the temporal weighting method, the temporal relationship between the candidate terms and the query terms are also considered. Therefore, we also ranged the values of *j* and *n* between the same values as the Experiment A.1. The penalization applied to the candidate terms without a temporal relationship with the query terms was 0.5 (see Formula 5.6). The other candidate terms do not have any penalization.

Considering the baselines *noQE100* and *Bo1QE100*, we verified that the temporal weighting method (TmpW) has improved the effectiveness, as shown in Table 6.6. The best values of each column appear in bold. All the obtained results are better than *noQE100*. The gains of the TmpW method are around 5%-14%. Precision@10 and MAP achieved the lowest and the highest gain, respectively.

	#Relevant retrieved	**MAP**	**P@10**	**P@15**	**P@20**
noQE100	4450	0.322	0.484	0.438	0.407
Query Expansion, *n=10*					
Bo1QE100	4824	0.361	0.498	0.471	0.449
Temporal Weighting					
(*) j=1	4826	0.362	0.502	0.473	0.451
(*) j=2	**4827**	**0.363**	0.505	0.477	**0.452**
(*) j=3	4812	0.362	0.504	0.**479**	**0.452**
j=4	4808	0.360	0.496	0.475	0.447
j=5	4799	0.360	0.505	0.474	0.448
j=6	4801	0.359	0.508	0.478	0.446
j=7	4788	0.357	**0.510**	0.475	0.446
j=8	4788	0.357	0.506	0.473	0.445
j=9	4760	0.354	0.506	0.473	0.443
j=10	4742	0.354	0.508	0.476	0.443
Query Expansion, *n=20*					
Bo1QE100	4853	0.365	0.501	0.476	0.452
Temporal Weighting					
(*) j=1	4852	**0.367**	0.505	0.481	**0.457**
(*) j=2	**4860**	**0.367**	0.503	**0.486**	0.459
(*) j=3	**4860**	**0.367**	0.508	0.485	**0.461**
(*) j=4	4857	**0.367**	0.506	0.481	0.456
j=5	4828	0.364	0.507	0.481	0.456
j=6	4824	0.363	0.504	0.479	0.454
j=7	4813	0.361	0.506	0.479	0.449
j=8	4808	0.360	0.506	0.478	0.449
j=9	4776	0.358	0.508	0.476	0.450
j=10	4770	0.358	**0.509**	0.477	0.450

Table 6.6. Results of temporal weighting with 100 topics.

Note that all the values of Precision@10 and Precision@15 are better than the two baselines. In the configuration *n=10;j=4*, Precision@10 is not above, but its value is very close to the reference value (*Bo1QE100*).

All the results marked with *(*)* are greater than *Bo1QE100* with gains around 0.5%-2.4%. The values of precision achieved the best gains, while MAP achieved only residual differences around 0.5-1%. These results were obtained with *j≤3* and *j≤4* when the system was configured with *n=10* and *n=20*, respectively.

We verified that the two query expansion methods, TmpW and *Bo1QE100* obtained better results when the number of terms of the reformulated query was configured with 20 (*n=20*), although with a residual difference when compared with the results obtained with *n=10*, around 0.5%-1.4%. However, the best result of Precision@15 was obtained by the TmpW method with *n=10.*

We observed that the best MAP (0.363) was obtained with a small value of *j* (*j=2*), such as in the temporal filtering method. On the other hand, temporal weighting achieved a better precision, when *j* is increased. The explanation of this is that the terms positioned on top of the ranking become the terms in which the number of associated *chronons* in common with the query terms is higher.

Experiment A.3 – Temporal aware Query Reweighting

The evaluation of the temporal aware query reweighting (TmpR) method focused on the variation of parameters β, and δ (see Formula 5.7). α was set to 1, given the most importance to the initial query terms. β and δ were ranged between 0.1 and 0.9, considering that $\beta+\delta\leq 1$. Due to the huge number of possible experiments, considering the parameters to change, this parameter had an increment of 0.1. Even so, the value of *j* was ranged between 1 and 5. This value was limited to 5, because in the experiments A.1, and A.2 we verified that the best results were obtained with *j<5*.

For this experiment, we only set *n* to 10. The two main reasons are the results obtained by the other tmpQE methods with *n=20* shown the same behavior than the model Bo1, and all the results are better.

Since the number of changed parameters is high in the experiments carried out with the TmpR method, Table 6.7 only shows the 10 best results obtained with the best values of MAP. The table also shows the results of the baselines *noQE100* and *Bo1QE100*. The best values of each column appear in bold.

We verified that all the results of the TmpR method are above the baseline *noQE100* with gains about 3%-13%. Precision@10 and MAP achieved the lowest and the highest gain, respectively, such as the other tmpQE methods.

Note that in the top 1000 retrieved documents, the number of relevant documents is significantly higher (4868). This number was obtained with values of MAP and precision, which are also higher or similar than the query expansion baseline *Bo1QE100*.

The values of MAP and precision obtained by the TmpR method that are simultaneously greater than the results of *Bo1QE100* are marked with *(*)* in Table 6.7. However, the gains achieved are around 0.4%-2%.

The 10 best results were obtained with the following parameters: β={0.4,0.5,0.6}, δ={0.2,0.3,0.4}, and j={4,5}. This means that the candidate terms associated with 4 or 5 *chronons* (j) of the query terms were weighted with a value near the maximum value, since $\beta + \delta \leq 1$. In fact, the best value of MAP (0.363) was achieved when the weight of those terms (β+δ) are 0.7, 0.8, and 0.9. We observed that the precision is lower when the parameter δ is higher.

We verified that the TmpR method also improved the effectiveness. The results obtained by TmpR are similar than the other tmpQE methods, although this method found more relevant documents for the 100 queries. The best results were obtained with a value of j greater than the other methods ($j \in [4,5]$). We saw that documents have an average of 4 *chronons* (see Section 4.3.1).

	#Relevant retrieved	**MAP**	**P@10**	**P@15**	**P@20**
noQE100	4450	0.322	0.484	0.438	0.407
Query Expansion, *n=10*					
Bo1QE100	4824	0.361	0.498	0.471	0.449
Temporal aware Query Reweighting					
β=0.4;δ=0.2; j=4	4823	0.362	0.498	0.473	0.453
(*) β=0.4;δ=0.3; j=4	4841	**0.363**	**0.500**	0.475	0.450
(*) β=0.4;δ=0.4; j=5	4851	**0.363**	0.499	**0.480**	0.447
(*) β=0.5;δ=0.2; j=4	4844	**0.363**	0.498	0.475	**0.455**
β=0.5;δ=0.2; j=5	4839	0.362	0.498	0.469	0.453
β=0.5;δ=0.3; j=4	4861	**0.363**	0.497	0.476	0.447
β=0.5;δ=0.3; j=5	4857	0.362	0.493	0.473	0.450
β=0.5;δ=0.4; j=4	4865	**0.363**	0.492	0.475	0.445
β=0.6;δ=0.2; j=4	**4868**	0.362	0.491	0.471	0.450
β=0.6;δ=0.3; j=4	4865	0.362	0.489	0.472	0.447

Table 6.7. Results of temporal aware query reweighting with 100 topics.

Experiments A – Comparative Analysis

In order to compare our three tmpQE methods, we also computed the robustness taking as reference the two baselines. Robustness(1) and Robustness(2), presented in Table 6.8, were computed by reference to *noQE100* and *Bo1QE100*, respectively. Robustness(2) was considered to compare our tmpQE methods with other query expansion method. This table also shows the best values of effectiveness obtained by each of our tmpQE methods, considering the MAP metric (see the three previous tables). The best values of each column appear in bold.

We verified that the greatest number of relevant documents (4841) in the top 1000 documents of the result set was found by the TmpR method. This method also obtained the best MAP (0.363) and P@15 (0.480). Note that although the value of the Robustness(1) is near than the value obtained by *Bo1QE100*, the TmpR method improved 54 queries when compared with *Bo1QE100*, achieving the best Robustness(2). This means that this method improves the average precision, promoting the relevant documents in the ranking.

The other two tmpQE methods achieved similar results of effectiveness and robustness. They obtained better values of precision and Robustness(1) than the temporal aware query reweighting method.

We conclude that our tmpQE methods improved both effectiveness and robustness, with similar results. Comparing with the results of the other query expansion method, *Bo1QE100*, our methods improved the average precision of more 7% of queries (76), and Precision achieved the best gains (around 0.5%-2.5%).

		Effectiveness				**Robustness (1)**			**Robustness (2)**		
	#Relevant retrieved	**MAP**	**P@10**	**P@15**	**P@20**	**RI**	**IMP**	**PQ**	**RI**	**IMP**	**PQ**
noQE100	4450	0.322	0.484	0.438	0.407						
Query Expansion, ***n=10***											
Bo1QE100	4824	0.361	0.498	0.471	0.449	0.400	69	29			
TmpF	4823	0.362	0.501	0.477	**0.452**	**0.520**	76	24	0.040	47	43
TmpW	4827	**0.363**	**0.505**	0.477	**0.452**	**0.520**	76	24	0.030	47	44
TmpR *β=0.4;δ=0.4*	**4841**	**0.363**	0.499	**0.480**	0.447	0.420	71	29	**0.100**	54	44

Table 6.8. The best results obtained by each tmpQE method with 100 topics.

6.4.3 Experiments B (40 topics)

The experiments presented in this subsection were performed to know how the words temporality of documents could improve the time-sensitive queries.

In Experiments B, the parameter *n* was set to 10, and *j* was ranged between 1 and 10.

Since the number of topics used in Experiments B is different from Experiments A, the retrieval system for the 40 topics of *TopicsSet2* should only find 1,975 relevant documents of the 40,000 retrieved documents.

We discuss the results obtained with this configuration in each of these experiments, which used the 40 time-sensitive topics of the CHAVE collection (*TopicsSet2*). Then, we compare the best results obtained by each tmpQE method.

Experiment B.1 – Temporal Filtering

Table 6.9 shows the results obtained by the temporal filtering method, and the baselines *noQE40*, and *Bo1QE40*. The best values of each column appear in bold. The TmpF method obtained good results, proving the effectiveness improvement in time-sensitive queries. The best values were obtained when $j \leq 4$.

We verified that all the results are above the baseline *noQE40* with considerable gains, between 6% and 14%. Such as in Experiments A, Precision@10 achieved the lowest gain. The highest gain was obtained by Precision@20.

The results obtained with $2 \leq j \leq 4$ are marked with (*), since they are all above the results of *Bo1QE40*. When $j=1$, only the Precision@10 is below than the value of *Bo1QE40*. The TmpF method achieved gains around 2%-7%. Precision@10 and Precision@15 achieved the lowest and the highest gains, respectively. However, the number of retrieved documents is a little lower than 1779, the value obtained by *Bo1QE40*.

Note that the best MAP (0.395), Precision@15 (0.460), and Precision@20 (0.436) were obtained when at least 3 *chronons* of pseudo-relevant documents $(j=3)$ matching the *queryTerms_chronons* set. The parameter *j* is higher than the value obtained by the same method when was applied to *TopicsSet1*.

	#Relevant retrieved	MAP	P@10	P@15	P@20
noQE40	1662	0.352	0.458	0.410	0.384
Query Expansion, *n=10*					
Bo1QE40	1779	0.387	0.473	0.430	0.419
Temporal Filtering					
j=1	1770	0.388	0.470	0.438	0.424
(*) j=2	**1776**	0.392	**0.483**	0.453	0.428
(*) j=3	1773	**0.395**	0.480	**0.460**	**0.436**
(*) j=4	1772	0.392	0.480	0.453	0.435
j=5	1767	0.381	0.473	0.442	0.418
j=6	1758	0.381	0.478	0.448	0.411
j=7	1751	0.374	0.475	0.437	0.409
j=8	1751	0.371	0.465	0.438	0.404
j=9	1740	0.373	0.470	0.435	0.395
j=10	1696	0.361	0.465	0.425	0.395

Table 6.9. Results of temporal filtering with 40 topics.

Experiment B.2 – Temporal Weighting

In the evaluation of the temporal weighting method, the penalization applied to the candidate terms without a temporal relationship with the query terms was 0.5 (see Formula 5.6). The other candidate terms do not have any penalization. This configuration is the same used in Experiment A.2.

Table 6.10 shows the reference results given by *noQE40*, and *Bo1QE40* and the results obtained by the temporal filtering method. The best values of each column appear in bold. The TmpW method improved the effectiveness (2%-12%) in time-sensitive queries. The best values were obtained when $j \leq 5$.

Such as in Experiments A, all the results are above the baseline *noQE40*. The TmpW method achieved considerable gains between 6%-12%. Precision@10 achieved the lowest. The highest was obtained by the metrics MAP and Precision@20.

Note that all the values of Precision@15 are better than the two baselines. All the results marked with *(*)*, were obtained with $2 \leq j \leq 3$, are greater than the results of *Bo1QE40*. The gains achieved by the TmpW method are between 2% and 6%. MAP and Precision@15 obtained the lowest and highest gains, respectively. However, the maximum number of retrieved documents is equal than the value obtained by *Bo1QE40*.

Note that the values of MAP are greater than the reference value when $j \leq 5$. All the values of the Precision@15 are above of the reference value. Precision@10 is only below than this value when *j=1*. Precision@20 is below for the configuration *j≥ 6*. So the best values of each metrics was obtained when *1≤j≤ 5*.

	#Relevant retrieved	MAP	P@10	P@15	P@20
noQE40	1662	0.352	0.458	0.410	0.384
Query Expansion, *n=10*					
Bo1QE40	1779	0.387	0.473	0.430	0.419
Temporal Weighting					
j=1	1770	0.388	0.470	0.438	0.424
(*) j=2	**1779**	**0.393**	0.483	**0.453**	**0.429**
(*) j=3	1764	**0.393**	0.483	0.450	0.426
j=4	1777	0.391	0.473	0.445	0.419
j=5	1776	0.390	0.473	0.438	0.420
j=6	**1779**	0.387	**0.485**	0.448	0.413
j=7	1770	0.385	0.483	0.443	0.413
j=8	1775	0.385	0.480	0.445	0.413
j=9	1767	0.385	0.478	0.443	0.413
j=10	1751	0.380	0.475	0.447	0.411

Table 6.10. Results of temporal weighting with 40 topics.

Experiment B.3 – Temporal aware Query Reweighting

Besides the parameters *n* and *j*, this experiment needs the configuration of more three: *α*, *β* and *δ* (see Formula 5.7), which are the same as Experiment A.3. *α* was set to 1, *β* and *δ* were ranged between 0.1 and 0.9, considering that *β*+*δ*≤ *1*.

The temporal aware query reweighting method improved the effectiveness in time-sensitive queries, achieving considerable gains. Table 6.11 shows the ten best results, considering MAP. All these values are above the baselines *noQE40* and *Bo1QE40*. These values were obtained with the following parameters: *β*={0.1,0.2,0.3}, *δ*={0.5,0.6,0.7,0.8,0.9}, and *j*=4. The best values of each column appear in bold.

The gains obtained by reference of *noQE40* are around 6%-13%. Precision@10 and MAP achieved the lowest and the highest gains, respectively, such as the other tmpQE methods.

Precision@10 and Precision@15 achieved the lowest (2%) and the highest (4%) gains, respectively, obtained by reference of *Bo1QE40*.

	#Relevant retrieved	MAP	P@10	P@15	P@20
noQE40	1662	0.352	0.458	0.410	0.384
Query Expansion, *n=10*					
Bo1QE40	1779	0.387	0.473	0.430	0.419
Temporal aware Query Reweighting, *j=4*					
β=0.1;δ=0.7	1791	0.396	0.470	0.433	0.426
β=0.1;δ=0.8	1787	0.396	0.473	0.437	**0.430**
β=0.1;δ=0.9	1787	0.396	0.473	0.433	0.429
β=0.2;δ=0.5	1788	0.396	**0.483**	0.443	0.428
β=0.2;δ=0.6	1792	0.396	0.480	0.443	0.423
β=0.2;δ=0.7	1794	0.397	0.478	0.443	0.426
(*) β=0.2;δ=0.8	**1796**	**0.398**	0.478	0.443	0.428
β=0.3;δ=0.5	1793	0.396	0.475	0.443	0.426
β=0.3;δ=0.6	1795	0.397	0.480	**0.445**	0.425
β=0.3;δ=0.7	1795	0.396	0.475	**0.445**	0.425

Table 6.11. Results of temporal aware query reweighting with 40 topics.

Note that the TmpR method found more 17 relevant documents than the other query expansion method *Bo1QE40,* with *β=0.2;δ=0.8*. The highest value of MAP (0.398) was obtained with the same configuration. This configuration is marked with (*) in Table 6.11.

Since α=1 and $\beta+\delta$=1, the terms with the same temporality as the query terms obtained the same weight as the initial query terms (1), while the weight assigned to the other new terms was 0.2; a great difference of weights between the new terms. So, we can conclude that the terms with the same temporality as the query terms obtained the same importance than the initial query terms in the reformulated query.

We verified that the temporal aware query reweighting method achieved the best MAP (0.398) and found more relevant documents (1796) than the other tmpQE methods. Note that the parameter *j* must be set to a higher value (*j=4*) to achieve the best results.

Experiments B – Comparative Analysis

Table 6.12 shows the best effectiveness obtained by each of our tmpQE methods, considering the MAP metric (see the three previous tables). The best values of each column appear in bold. In order to better compare these methods, this table also shows their robustness, Robustness(1) and Robustness(2), computed by reference to *noQE40* and *Bo1QE40*, respectively, such as in the other experiments.

		Effectiveness				**Robustness (1)**			**Robustness (2)**		
	#Relevant retrieved	**MAP**	**P@10**	**P@15**	**P@20**	**RI**	**IMP**	**PQ**	**RI**	**IMP**	**PQ**
noQE40	1662	0.352	0.458	0.410	0.384						
Query Expansion, *n=10*											
Bo1QE40	1779	0.387	0.473	0.430	0.419	0.500	29	9			
TmpF	1773	0.395	0.480	**0.460**	**0.436**	**0.575**	31	8	0.350	22	8
TmpW	1779	0.393	**0.483**	0.453	0.429	**0.575**	31	8	0.325	22	9
TmpR *β=0.2;δ=0.8*	**1796**	**0.398**	0.478	0.443	0.428	0.550	31	9	**0.425**	28	11

Table 6.12. Best results obtained by each tmpQE method with 40 topics.

We verified that the temporal aware query reweighting method achieved the best MAP (0.398) and found the greatest number of relevant documents (1796) in the top 1000 documents. However, the precision was improved by the other methods. Temporal weighting obtained the best Precision@10 (0.483). Temporal filtering obtained the best values of Precision@15 (0.460) and Precision@20 (0.436).

Our methods obtained a better robustness than the other query expansion method (*Bo1QE40*), improving more queries and penalizing less. The Robustness(1) is similar in our methods.

Considering *Bo1QE40* as reference, the temporal aware query reweighting method obtained the best robustness (0.425), improving 28 queries, although it penalizes more queries than the other methods. Temporal filtering and temporal weighting methods obtained a similar Robustness(2) (0.575).

We verified that our tmpQE methods significantly improved both effectiveness and robustness in time-sensitive queries. Although, the difference between the number of improved queries by our methods and the other query expansion method *Bo1QE40* are only 2 queries (5%), our methods achieved a better average precision in 28 of 40 queries (Robustness(2)). This means that our methods can promote the relevant documents in the ranking. Precision@15 achieved the best gain (7%) with the temporal filtering method, considering *Bo1QE40*.

6.4.4 Experiments C (18 topics)

Experiments C were performed to analyze how explicit temporal information of the queries changes the results, since the tmpQE methods already process the implicit temporal information of such queries.

These experiments used all the time-sensitive topics of the CHAVE collection, but with explicit temporal information in their text (*TopicsSet3*). This group of topics is a sub-set of *TopicsSet2*, employed in Experiments B.

Experiments C were divided into two types: C.1, where the processing only took into account the implicit temporal information of the queries; C.2, where both explicit and implicit temporal information were used in the query expansion.

In Experiments C.2, the processing of a given query includes its explicit temporal information found in its text. For this reason, it was necessary to create another set of *chronons* – *query_ExplicitChronons* to store this information.

We followed two different strategies for processing of the 18 topics of *TopicsSet3*. First, we added the *chronons* found in the query text (*query_ExplicitChronons*) to the set of *chronons*, which were associated to the query terms (*queryTerms_chronons*). We observed that these new *chronons* were already members of this set in the all topics of *TopicsSet3*. For that reason, the results did not change.

The second strategy is based on the *chronons* restriction of the *queryTerms_chronons* set to the time period defined by the explicit temporal information of the query. For example, for the query C350, considering the following initial *queryTerms_chronons* set:

```
queryTerms_chronons ={1994-10-23,1995-05-01,1994-10-22,1994-05-04,
                          1994-05-05,1994-05-03,1982-XX-XX,1994-XX-XX}
```

and the explicit *chronons* found in the text of the query C350:

```
query_ExplicitChronons={1994-XX-XX}
```

As we observed that the year of the two *chronons 1995-05-01* and *1982-XX-XX* is not *1994*, they are excluded from the set. So, the *queryTerms_chronons* is updated to:

```
queryTerms_chronons={1994-10-23,1994-10-22,1994-05-04,1994-05-05,
                                               1994-05-03,1994-XX-XX}
```

Although the Experiments C.2 followed the two different strategies, we only present the results obtained when the second strategy was applied, since the first strategy and the Experiments C.1 obtained the same results.

As well as in the other experiments, we analyzed not only the effectiveness of the set of the queries but also its robustness, taking as reference the two baselines. Robustness (1) was computed by reference to the baseline *noQE18*, without query expansion. In order to compare our methods with other query expansion method, we computed Robustness (2) considering *Bo1QE18*.

In all the experiments carried out with *TopicsSet3*, the number of terms (*n*) was set to 10, and the parameter *j* was ranged between 1 and 5. For the evaluation of the temporal aware query reweighting method, we used the same values of Experiment A.3 and Experiment B.3. So, the parameter β and δ ranged between 0.1 and 0.9, considering $\beta+\delta\leq 1$ (see Formula 5.7).

As reference, we know that in the retrieval of these 18 topics, the result set must be composed of 18,000 documents, from which 581 should be relevant documents.

Given the considerable number of results obtained, it was not possible to show all of them. So, Table 6.13 only shows the best results obtained for each tmpQE method, besides the baselines *noQE18* and *Bo1QE18*. The best values of each column appear in bold. We verified that all the results obtained by our tmpQE methods are above of the two baselines.

Experiment C.1

Our tmpQE methods improved both effectiveness and robustness in the time-sensitive queries of *TopicsSet3*, considering both baselines *noQE18* and *Bo1QE18*.

The temporal aware query reweighting method achieved gains in effectiveness around 6%-15%, considering the baseline *noQE18*. The gains obtained by the other two tmpQE methods were lower, around 5%-11%. In the three tmpQE methods, MAP and Precision@10 achieved the lowest and the highest gain, respectively.

Our tmpQE methods also achieved better results than the other query expansion method, *Bo1QE18*. The lowest gain of effectiveness obtained by these methods verified in MAP. In the temporal filtering method, MAP achieved 2% and Precision@20 obtained the highest gain, around 6%. Temporal weighting achieved 3% in MAP. Precision@15 obtained the highest gain, 5%. The other method obtained the best gains, around 3% and 8%, in MAP and Precision@10, respectively.

The best results of the temporal aware query reweighting method were obtained with *j=4*, while for the other methods it was with *j=3*.

Temporal filtering is the method that provides the highest improvement in robustness. It improved more 4 queries than the other query expansion method *Bo1QE18*, taking as reference the results of *noQE18*. Besides that, this method still managed to improve 10 queries, considering the results obtained by *Bo1QE18*.

Experiment C.2

As in Experiments C.1, our tmpQE methods also improved both effectiveness and robustness in the time-sensitive queries of *TopicsSet3*, both baselines *noQE18* and *Bo1QE18*.

We observed that in all the experiments that considered the explicit temporal information of queries, the best values were achieved when *j=1*. This means that only one *chronon* is required to the matching between the set of the *chronons* in both query and a given term. Since there is an explicit restriction to the time defined in the query, more restrictions provides an exclusion of too many candidate terms that would be important for the query.

In Experiments C.2, our tmpQE methods obtained better results in all the metrics of effectiveness. The method with the best effectiveness was also the temporal aware query reweighting. The gains are around 7%-16% and 5%-11%, taking as reference the baseline without query expansion and the other query expansion method, *noQE18* and *Bo1QE18*, respectively. MAP and Precision@10 achieved the lowest (7%) and the highest gains (16%) relating to *noQE18*. Considering the other baseline, *Bo1QE18*, MAP and Precision@20 obtained the lowest (5%) and the highest (11%) gains, respectively.

Table 6.13 shows the results obtained by the temporal aware query reweighting method with two distinct configurations. The first one (*β=0.5;δ=0.4*) is the same configuration that obtained the best results in Experiments C.1. This configuration also obtained the best results in Experiments C.2. The other one (*β=0.1;δ=0.6*) with a bigger difference between the *β* and *δ* weights only obtained the best MAP and Precision@15.

MAP and Precision@15 are the metrics with the lowest and the highest gains obtained by the other tmpQE methods. Precision@15 achieved the same gains by these two methods, approximately 13% and 8%, considering *noQE18* and *Bo1QE18*, respectively. The MAP of the temporal filtering method is a little more than the other method, but it is not a significant difference. Considering *noQE18* and *Bo1QE18*, 6% and 4% are the gains obtained by the filtering method, and 5% and 3% are the gains of the other method.

The robustness of our tmpQE methods has little differences, the improvement of one/two queries. However, the temporal filtering achieved the best Robustness (1), 61.1%, taking as reference the baselines *noQE18*. The temporal aware query reweighting method improved the highest number of queries (12), achieving the best Robustness (2), 44.4%, considering the results of *Bo1QE18*.

Experiments C – Comparative Analysis

We verified that our tmpQE methods have led to a big increase of the effectiveness of the *TopicsSet3*, both in MAP and precision, although they only found more 1-2% of relevant retrieved documents than the other query expansion method (*Bo1QE18*). Since the precision is also better, we can conclude that the relevant documents were better positioned in the ranking.

Table 6.13 shows the best results obtained in the experiments with 18 topics. The best values of each column appear in bold. In Experiments C.2, all the best results were obtained with *j=1*.

We observed that the number of improved queries is the highest when the explicit temporal information was not considered by our tmpQE methods, although with minimal differences. However, the effectiveness was increased when that information was used in the query expansion processing (Experiments C.2). This means a significant increase of the effectiveness in the queries in which an improvement was given by the explicit temporal information.

The best results of effectiveness were obtained by the temporal aware query reweighting method when the explicit temporal of queries is considered (Experiments C.2). It achieved a MAP of 0.462 for a Precision@10 of 0.517. The methods of temporal filtering and temporal weighting achieved a similar performance in terms of effectiveness.

The robustness obtained similar values between the three tmpQE methods in the two types of experiments (C.1 and C.2). The best values of robustness were achieved in Experiments C.1. The temporal filtering method achieved the best values, 66.7% and 44.4%. This method improved the average precision in 15 queries and worsened in only 3 of the 18 queries, taking as reference the baseline *noQE18*. Considering the values of *Bo1QE18*, this method improved 10 queries, penalizing only 2 queries. However, temporal aware query reweighting improved the highest number of queries (12), achieving also the best Robustness (2), 44.4%.

We observed that the differences between the results obtained in Experiments C.1 and Experiments C.2 are not very marked, although Experiments C.2 obtained better results in both MAP and precision. Therefore, the explicit temporal information of the text queries is important to restrict the temporal scope of the words. However, the results obtained with our collection do not show greater differences, since the temporal scope of documents timestamp is only two years [1994-1995]. The limited number of queries can also influence the results. Furthermore, mostly of temporal information, which is implicit in documents content and explicit in text queries, is in the same temporal scope.

	#Relevant	Effectiveness				Robustness (1)			Robustness (2)		
	retrieved	**MAP**	**P@10**	**P@15**	**P@20**	**RI**	**IMP**	**PQ**	**RI**	**IMP**	**PQ**
noQE18	525	0.431	0.444	0.411	0.372						
Query Expansion, *n=10*											
Bo1QE18	533	0.441	0.472	0.430	0.386	0.333	11	5			
Temporal Filtering											
(C.1) implicit; *j=3*	**539**	0.451	0.494	0.452	0.408	**0.667**	15	3	**0.444**	10	2
(C.2) implicit + explicit	536	**0.456**	**0.500**	**0.463**	**0.411**	0.611	14	3	0.333	9	3
Temporal Weighting											
(C.1) implicit; *j=3*	**540**	**0.454**	0.494	0.452	0.406	0.500	13	4	0.111	8	3
(C.2) implicit + explicit	538	0.453	**0.500**	**0.463**	**0.414**	**0.556**	14	4	**0.167**	8	5
Temporal aware Query Reweighting											
(C.1) implicit; *j=4*											
β=0.5;δ=0.4	538	0.455	0.511	0.448	0.417	**0.444**	13	5	0.222	10	6
(C.2) implicit + explicit											
β=0.5;δ=0.4	**539**	0.457	**0.517**	0.444	**0.428**	**0.444**	13	5	**0.444**	12	4
β=0.1;δ=0.6	536	**0.462**	0.511	**0.452**	0.419	0.389	12	5	0.222	9	5

Table 6.13. Results of the experiments with 18 topics.

6.4.5 Experiments D (60 topics)

We performed a set of experiments to verify if the temporal relationship between words also improves the effectiveness of time-insensitive queries. The experiments were carried out with the 60 time-insensitive topics of the CHAVE collection (*TopicsSet4*).

The values of parameters configured in Experiments D are the same used in the other experiments. The number of terms (*n*) was set to 10, and the number of matching *chronons* (*j*) ranged between 1 and 10. The parameters used in the temporal aware query reweighting method (β and δ) ranged between 0.1 and 0.9, considering $\beta+\delta\leq 1$ (see Formula 5.7).

The result set of the retrieval for the 60 topics should be composed of 3,606 relevant documents of the 60,000 retrieved documents.

Table 6.14 shows the best values of effectiveness and robustness achieved by each of our tmpQE methods, and the two baselines, *noQE60* and *Bo1QE60*. The best values of each column appear in bold. We verified that the best values were obtained when *j=1*, which means that a term is considered more relevant when it has just one *chronon* associated that matches the *queryTerms_chronons* set. However, the temporal aware query reweighting method achieved the best robustness when *j=4*, although the best effectivess was obtained with *j=1*.

		Effectiveness				**Robustness (1)**			**Robustness (2)**		
	#Relevant retrieved	**MAP**	**P@10**	**P@15**	**P@20**	**RI**	**IMP**	**PQ**	**RI**	**IMP**	**PQ**
noQE60	2788	0.302	0.502	0.457	0.422						
Query Expansion, *n=10*											
Bo1QE60	3045	0.344	0.515	**0.498**	0.469	0.333	40	20			
TmpF, *j=1*	3056	0.344	**0.523**	0.493	0.468	**0.500**	45	15	**0.033**	29	27
TmpW, *j=1*	3056	0.344	**0.523**	0.496	0.469	**0.500**	45	15	0.017	29	28
TmpR											
β=0.4;δ=0.3;j=1	**3089**	**0.346**	0.512	0.495	0.468	0.300	39	21	-0.267	22	38
β=0.5;δ=0.2;j=5	3053	0.345	0.518	0.496	**0.475**	0.400	42	18	-0.200	24	36

Table 6.14. Best results obtained with time-insensitive queries.

We verified that all the results by our tmpQE methods are above of *noQE60*. MAP and Precision@10 are the metrics with the highest and the lowest gains obtained by these methods, such as in the other experiments. The methods of temporal filtering and temporal weighting obtained the same results, with a little difference in Precision@15. The gains achieved by these two methods are between 4% and 14%. The temporal aware query reweighting method obtained gains around 2%-15%.

The temporal aware query reweighting is the only method that improves MAP, considering *Bo1QE60*, although with residual gains (around 0.5%). This one is the tmpQE method, which found the most number of relevant documents (3089), more around 1.5% than the baseline *Bo1QE60*. However, this tmpQE method obtained the worst robustness, penalizing more queries than improving, taking as reference the same baseline.

Taking the same reference values, the other two methods improved Precision@10 (around 2%) and robustness (Robustness(2)), improving almost 50% of the queries (29). However, the value of Robustness(2) is not high, since they penalized almost the same number of queries. The difference is only 1-2 queries. On the other hand, these methods improved more 5 queries than *Bo1QE60*, achieving the best Robusteness(1), taking *noQE60* as reference.

So, we verified that the effectiveness of our tmpQE methods is very similar, with little differences between them. The temporal aware query reweighting method achieved a better MAP and found more relevant documents than the other two methods. On the other hand, the other two methods obtained the best Precision@10 and robustnesss, comparing with the two baselines.

Our tmpQE methods improved both effectiveness and robustness of the time-insensitive queries. However, the differences of the results obtained by these methods and the other query expansion method are not significant. Our intuition is that if we combine the temporal approach with the words occurrences, this type of queries can also be improved with greater gains. We intend to explore this idea in the near future.

6.5 Conclusions

This chapter describes the workbench created in the Terrier platform for the effectiveness evaluation of our temporal methods for query expansion. The CHAVE collection was used in all the performed experiments. The first sections explain the platform and the implementation details of the five tmpQE methods (see Section 5.3).

The experiments were designed in order to verify how important the words temporality used by these methods in the retrieval process is, considering any query, both time-sensitive and time-insensitive. Experiments A use all the topics of the collection (time-sensitive and time-insensitive). In Experiments B, only the time-sensitive topics are considered. The set of topics employed in Experiments C is composed of the time-sensitive topics with explicit temporal information found in their title and description. A set of the time-insensitive topics is used in Experiments D.

The experiments carried out with two of the methods, temporal profile of terms and temporal profile of pseudo-relevant documents, are not enough to be included in the evaluation. So, the presented experiments were carried out with the other three methods. Temporal filtering removes the terms outside the temporal space of the original query terms from the set of candidate terms. Temporal weighting penalizes the score of terms that do not co-occur temporally with the original query terms, dropping their places in the selection ranking. Temporal aware query reweighting increases the weight of terms that are in the same temporal space of the original query terms, considering only the terms already selected to be included in the expanded query.

Due to the huge number of possible experiments, considering the parameters that can be changed, we had to set some of them, such as the number of pseudo-relevant to be considered, and the maximum number of terms used in the reformulated query. A detailed description of the parameters configuration is in Section 6.4.1. The definition of these and the variation of the other parameters take into account to work in the more restrictive conditions, in addition to the main objective of the experiments. For this reason, the results obtained by our tmpQE methods are not the best possible. An example is presented in Experiments A (see Section 6.4.2).

By the analysis of the obtained results, we verified that the query expansion brought benefits to the retrieval, since the effectiveness was improved in all experiments.

The results show that our tmpQE methods improved both effectiveness and robustness for time-sensitive and time-insensitive queries, although with greater evidence in time-sensitive queries in which these methods achieved higher gains (Experiments B and C). Our intuition is that if we combine the temporal approach with the words occurrences, time-insensitive queries can also be improved with greater gains. We intend to explore this idea in the near future.

One of the issues raised by the evaluation process is the small temporal scope of the CHAVE collection, only two years [1994-1995], which means that the words temporality is practically the same, as illustrated in Section 4.3.

Note that the best results were obtained by temporal aware query reweighting with ($\beta+\delta\approx 1$), in particular in Experiments B and C with time-sensitive topics. This means that the terms with the same temporality as the query terms obtained practically the same importance ($\beta+\delta=1$) than the initial query terms ($\alpha=1$) in the reformulated query (see Formula 5.7).

In general, our methods obtained very close results in all the experiments. The temporal filtering and temporal weighting methods had a similar performance, improving the precision. Temporal aware query reweighting obtained a better MAP, since more relevant documents were found and they were better positioned in the ranking. Temporal weighting obtained a lower robustness than the other methods.

Temporal weighting uses a penalization of the term score given by the temporal distance $td(t)$ between the temporal spaces of the original query terms and the candidate term (see Formula 5.6). In this work, $td(t)$ is computed following a discrete approach. In practice, the penalization is 0.5 when the temporal space of the candidate terms is outside the temporal space of the original query terms; otherwise 0. However, this penalization needs to be analyzed in more depth. It is our intention to explore a continuous approach for computing the temporal distance $td(t)$, promoting a great refine of the penalization.

Of course, this is a first study of the applicability of the words temporality in information retrieval. We verified that our models can improve the results, but more works need to be done to thoroughly explore the combination with the various parameters. An example, we intend to combine words temporality with words occurrence in the term-weighting formulas. Some other examples are also mentioned above. The next chapter describes the directions for future work.

Chapter 7

Conclusions and Future Work

The work developed in this thesis takes the advantage of the temporal relationships between words to improve the effectiveness of retrieval systems. This chapter summarizes the approaches proposed in order to incorporate temporal information in a retrieval model. The achievements made and some directions in which the work can be conducted in the near future are also presented.

Time is an important dimension in computer science area. For example, for text understanding, it allows the identification of relations between entities, facts or events. In the context of the user's information need, it is also a relevant element. In fact, it can benefit a large range of tasks. Thus, it is no surprise that its importance is increasing in information retrieval area.

The center of our research is to take advantage of temporal information found in the content of documents to improve effectiveness in information retrieval systems. Although there are some works with the same purpose, their models use a bag of words and a bag of temporal information without any dependence between these two sets, contrary to our work. To the best of our knowledge, our approach is the unique that is based on establishing of temporal relationships between words. These relationships are obtained by the temporal segmentation of texts, using the temporal information found in the content of documents. The temporality of a segment is shared by all of its words. In this way, the words temporality is defined and introduced in the retrieval models by the temporal indexes.

Our proposed approach of temporal information retrieval was applied using the query expansion technique, although it can be applied to other issues and components of retrieval systems. Therefore, this first application of our approach launches the foundations for further research on taking advantage of the temporal relationships between words to improve the effectiveness of retrieval systems.

The results obtained by our proposed methods show that the use of words temporality enhances the effectiveness and the robustness of retrieval systems. In the experiments performed with temporal filtering, temporal weighting and temporal aware query reweighting, we verified that these three temporal query expansion methods obtained very closed results, showing similar improving effects. Temporal filtering and temporal weighting had a similar performance, improving the precision. Temporal aware query reweighting obtained more relevant documents, promoting them in the ranking, and obtaining a better MAP. Temporal weighting was the method with the worst robustness.

In general, the improvement was obtained for both time-sensitive and time-insensitive queries, although with greater evidence in time-sensitive queries in which the temporal methods achieved higher gains, from 2% to 16%. Our intuition is that if we combine the temporal approach with the words occurrences, time-insensitive queries can also be improved with greater gains. We intend to explore this idea in the near future. We verified that the explicit temporal information found in the queries text is important to limit the temporal scope of the words, giving greater improvements, from 7% to 16%.

In order to obtain the temporal relationships between words, temporal information extraction and temporal segmentation must be performed in documents and queries. Temporal information occurs in texts as temporal expressions expressed in an explicit or implicit way. Once the extraction of such information is carried out, it is later used in the temporal segmentation, since this segmentation is based on the temporal discontinuities found in the text.

Our mother tongue was chosen to be used in this work. Since there was no available tools that can determines the temporal relationships between words in Portuguese texts, we also proposed some approaches for this purpose, namely to extract temporal information and to temporally segment Portuguese texts. Based on these approaches, we developed a toolset from scratch that is available at https://sites.google.com/site/olgacraveiro/home.

The proposed approach for extraction of temporal information uses classified temporal patterns, based on regular expressions to recognize the temporal expressions. The patterns are generated from the word co-occurrences obtained from Portuguese corpora and a pre-defined seed keywords set, which were derived from temporal references. The temporal expressions are annotated in the original text, and their corresponding normalized format is included in the annotation, where possible. Even with a set of limitations and simplifications, our approach has shown promising results in the effectiveness evaluation, a precision in the range of 78%-84% and a recall in the range of 64%-75%.

We also proposed an approach for temporal segmentation of texts in the Portuguese language. This approach uses the temporal information expressed in the content of documents or in the document metadata to divide the text into temporal coherent segments. The segments allow the establishment of a relationship between words and time. Therefore, the words can also be temporally related by using this relationship, since all the words of a segment share its temporality. Although this segmentation approach was used to obtain the temporality of words incorporated in retrieval model, it can be applied in other contexts. Indeed, our temporal segmentation approach was already applied in temporal focused Web crawling [Pereira et al., 2014, Pereira, 2013].

We also performed a temporal characterization of the two document collections in Portuguese, Second HAREM and CHAVE, which was focused on the relevant attributes for temporal-based text segmentation. Temporal information appears in the content of almost all documents of both collections. A deep analysis was performed on CHAVE, taking into consideration documents, topics and relevance judgments. Its usability for temporal *ad hoc* IR research was confirmed.

Although our work was only focused on effectiveness evaluation, leaving the efficiency problem for the next stage, we are aware that the system efficiency is an important issue. Given that it is for use in huge Web information retrieval applications, the need for a careful tradeoff between effectiveness and efficiency must be considered. Taking into account the application of our approaches in this context, we identify three main factors that must be treated with special care: document processing time, temporal index size and query execution time.

Document processing time includes temporal preprocessing, keyword extraction, and building of term and temporal indexes. Since the number of operations that must be executed until the temporal information is available in the retrieval system, the time spent of each one must be minimized.

Due to the large amount information storages in temporal indexes, this parameter must be studied, since the time spent in searching the indexes can compromise the efficiency of the query processing. Temporal indexes must storage the temporality of words. The large amount of this information requires that the storage must be carefully organized in order to store only the strictly required information to allow using the temporality of words and their positions in documents during the query processing, without compromising efficiency in obtaining the results. The use of storage techniques to improve efficiency must be considered, such as MapReduce indexing [McCreadie et al., 2012].

Note that although the focus of the evaluation carried out was the effectiveness, the efficiency was also a concern. It was taken into consideration in defining the structure of the temporal indexes and in the choice of hash tables as the data structures used in the developed software. In the temporal information extraction, the efficiency was also improved. Note that the execution time decreases about 27.5% by introducing the Step#1 in the annotation approach. The Step#1 excludes all the sentences that cannot have a temporal expression from the processing, which means that only sentences with date and time references and/or temporal words are processed.

In the following section, we outline some directions in which the work can be conducted in near future.

7.1 Future Work

The present work can be easily applied in various contexts, since time is an important dimension in document processing. Our approach of temporal retrieval preprocessing to give the temporal relationships between words is an important tool for future research. Although in the scope of this work, there is still a lot of research opportunities and, our intention for future work is to further explore the temporality of words in clustering and in document ranking, namely:

- combining the words temporality with the words occurrences. In this way, the weight of terms in documents depends on their occurrence in temporal segments. We believe that this combination can also improve time-insensitive queries with greater gains;
- implementing and evaluating the temporal operators, which can be used in queries with temporal restrictions;
- classifying queries in time-sensitive and time-insensitive, which is an important issue that also must be explored.

The code optimization was not a priority during the software development. Since this optimization can decrease the time processing, it is important to do this to enhance the efficiency of the developed applications. The efficiency in extraction, segmentation, and retrieval, is an important issue to study in near future.

There are also many possible extensions and improvements that can be performed in the proposed approaches. Some of them are described below.

Temporal Query Expansion Methods

Our temporal methods have been applied to Portuguese, which is our mother tongue, but as they are not dependent on language, it is perfectly possible to use them with other languages, such as English. The main drawback is in the obtaining the temporality of words, which is given by the temporal segmentation of texts. Thus, firstly, the segmentation approach must be adapted for the chosen language.

The formula introduced in temporal weighting method to penalize the score of candidate terms follows a discrete approach (see Formula 5.6). We want to explore also the continuous approach for computing temporal distance.

All the proposed methods can also obtain better results considering in their formulas, not only the temporality of words, but also, the words occurrence in the same temporal segments, as we mentioned above.

We were forced to put much of our effort in building the temporal preprocessing tools and therefore had less time to dedicate to retrieval. For this reason, although we have performed some experiments with temporal profile of terms and temporal profile of pseudo-relevant documents, they are not enough to be included in the thesis. The two methods require further study.

Temporal Extraction Approach

The extraction approach was defined to process texts in the Portuguese language and to be used in text segmentation. However, this approach can also be applied to foreign languages (English, for instance) and to another contexts, such as other kind of named entities recognition. For this purpose, it is required to define the lexical and grammatical markers by a deep study of the chosen language and the context, based on a careful statistical analysis, for example.

A research direction is the variation of used parameters, such as *n*, the number of maximum words on the temporal expression, and low frequency threshold. The variation of the *n* value must be carefully study, since increasing this parameter makes the COP module more complex. Additionally, the lexical and grammatical markers can be tuned and the pruning step resorting to stopwords to lower the rate of false positives must be improved.

At this stage, a manual validation is carried out to prune some incorrect patterns and to associate the correct semantic classification. However, we intend to apply a supervised algorithm for future work, since now we have considerable data to create a training collection.

The resolution of temporal expressions classified as duration or frequency can be provided, because they are not considered yet. The indexical references resolution must be improved, namely *anaphoric timexes* that are at this moment correctly resolved only if they contain a modifier marker.

Concerning the efficiency, this approach can also be further simplified, ignoring all temporal expressions that cannot be transformed into *chronons*, since the expressions will not be used in the temporal segmentation. This way, the efficiency in segmentation is also improved.

Temporal Segmentation Approach

Supported by a simple rule-based algorithm, despite the auspicious result of 0.15 for WD, the method can be improved using sentence as minimal segment size. Furthermore, the process of topic change detection may also be improved using for instance thesaurus and stemming.

The proposed approach only process texts in the Portuguese language. As we intend to explore the temporality of words in documents written in other language, such as English, it is requires an adaptation of the segmentation for processing texts in the chosen language. For this purpose, it is required a dictionary, a thesaurus and a stemming for that language, which is not difficult to obtain. However, the required collections to promote the evaluation can be more difficult to get.

The application of temporal segmentation algorithms for Web collections could be particularly interesting, as paragraphs or sentences involving anchor text that can be marked with the publication date of the linked page. Web pages can also be temporally segmented using these additional temporal references, even if without any temporal expressions in their content. Although the temporal segmentation was already applied in Web crawling, it can also be more explored in this context.

Additionally, the application of temporal text segmentation can be useful to the segmentation of text included in other media type, instead of documents.

Bibliography

[lin, nd] (n.d.). Linguateca homepage. http://www.linguateca.pt/ [September 20th, 2015].

[Ahn et al., 2005] Ahn, D., Adafre, S. F., and de Rijke, M. (2005). Extracting temporal information from open domain text: A comparative exploration. In *DIR 2005: 5th Dutch-Belgian Information Retrieval Workshop*, pages 3–10.

[Allen, 1983] Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Communications of ACM*, 26(11):832–843.

[Alonso, 2008] Alonso, O. (2008). *Temporal Information Retrieval.* PhD thesis, University of California, Davis - Department of Computer Science (CS UCDavis).

[Alonso et al., 2009a] Alonso, O., Baeza-Yates, R., and Gertz, M. (2009a). Effectiveness of temporal snippets. In *WSSP Workshop: Workshop on Web Search Result Summarization and Presentation WWW'09*, Madrid, Spain.

[Alonso et al., 2009b] Alonso, O., Gertz, M., and Baeza-Yates, R. (2009b). Clustering and exploring search results using timeline constructions. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 97–106, New York, NY, USA. ACM.

[Alonso et al., 2011] Alonso, O., Strötgen, J., Baeza-Yates, R., and Gertz, M. (2011). Temporal information retrieval: Challenges and opportunities. In *Proceedings of the 1st International Temporal Web Analytics Workshop (TWAW 2011)*, pages 1–8.

[Amati, 2003] Amati, G. (2003). *Probability Models for Information Retrieval based on Divergence from Randomness*. PhD thesis, Department of Computing Science, University of Glasgow.

[Amati and Van Rijsbergen, 2002] Amati, G. and Van Rijsbergen, C. J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389.

[Amodeo et al., 2011] Amodeo, G., Amati, G., and Gambosi, G. (2011). On relevance, time and query expansion. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM'11, pages 1973–1976, New York, NY, USA. ACM.

[Baeza-Yates, 2005] Baeza-Yates, R. (2005). Searching the future. In *SIGIR Workshop MF/IR*.

[Baeza-Yates and Ribeiro-Neto, 1999] Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. ACM Press / Addison-Wesley, Boston, MA, USA.

[Baptista, 2003a] Baptista, J. (2003a). Evaluation of finite-state lexical transducers of temporal adverbs for lexical analysis of Portuguese texts. In Mamede, N., Baptista, J., Trancoso, I., and das Graças Volpe-Nunes, M., editors, *Computational Processing of the Portuguese Language. Proceedings of the PROPOR'2003 - 6th Workshop on Computational Processing of the Portuguese Language - Written and Spoken*, pages 235–242. Berlin: Springer. Lecture Notes in Computer Science - Lecture Notes in Artificial Inteligence 2721.

[Baptista, 2003b] Baptista, J. (2003b). Some families of compound temporal adverbs in Portuguese. In *Workshop on Finite-State Methods for Natural Language Processing, International Conference of the European Chapter of the Association for Computational Linguistics*, pages 97–104.

[Beeferman et al., 1999]Beeferman, D., Berger, A., and Lafferty, J. (1999). Statistical models for text segmentation. *Mach. Learn.*, 34(1-3):177–210.

[Berberich and Bedathur, 2013] Berberich, K. and Bedathur, S. (2013). Temporal diversification of search results. In *SIGIR 2013 Workshop on Time-aware Information Access (TAIA'2013).*

[Berberich et al., 2007] Berberich, K., Bedathur, S., Neumann, T., and Weikum, G. (2007). A time machine for text search. In *Proceedings of the 30th Annual International ACM SIGIR*

Conference on Research and Development in Information Retrieval, SIGIR'07, pages 519–526, New York, NY, USA. ACM.

[Berberich et al., 2010] Berberich, K., Bedathur, S. J., Alonso, O., and Weikum, G. (2010). A language modeling approach for temporal information needs. In *ECIR*, pages 13–25.

[Bestgen and Vonk, 1995] Bestgen, Y. and Vonk, W. (1995). The role of temporal segmentation markers in discourse processing. *Discourse Processes*, 19(3):385–406.

[Bick, 2000] Bick, E. (2000). *THE PARSING SYSTEM "PALAVRAS" Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. PhD thesis, Department of Linguistics, University of rhus, DK.

[Bramsen et al., 2006] Bramsen, P., Deshpande, P., Lee, Y. K., and Barzilay, R. (2006). Finding temporal order in discharge summaries. In *AMIA '06: Proceedings of the American Medical Informatics Association Annual Symposium*, pages 81–85, Washington DC; USA.

[Brin and Page, 1998] Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. In *Computer Networks and ISDN Systems*, pages 107–117. Elsevier Science Publishers B. V.

[Brucato and Montesi, 2014] Brucato, M. and Montesi, D. (2014). Metric spaces for temporal information retrieval. In *Proceedings of the 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014*, pages 385–397.

[Campos et al., 2014a] Campos, R., Dias, G., Jorge, A. M., and Jatowt, A. (2014a). Survey of temporal information retrieval and related applications. *ACM Comput. Surv.*, 47(2):15:1–15:41.

[Campos et al., 2014b] Campos, R., Dias, G., Jorge, A. M., and Nunes, C. (2014b). GTE-Rank: Searching for implicit temporal query results. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 2081–2083.

[Carletta, 1996] Carletta, J. (1996). Assessing agreement on classification tasks: the kappa statistic. *Comput. Linguist.*, 22:249–254.

[Carpineto et al., 2001] Carpineto, C., de Mori, R., Romano, G., and Bigi, B. (2001). An information-theoretic approach to automatic query expansion. *ACM Trans. Inf. Syst.*, 19(1):1–27.

[Carpineto and Romano, 2012] Carpineto, C. and Romano, G. (2012). A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1.

[Chauvenet, 1960] Chauvenet, W. (1960). *A Manual of Spherical and Practical Astronomy V. II. 1863. Reprint of 1891*. 5th ed. Dover, N.Y.

[Costa and Branco, 2012] Costa, F. and Branco, A. (2012). Extracting temporal information from Portuguese texts. In de Medeiros Caseli, H., Villavicencio, A., Teixeira, A. J. S., and Perdigão, F., editors, *PROPOR*, volume 7243 of *Lecture Notes in Computer Science*, pages 99–105. Springer.

[Craveiro et al., 2008] Craveiro, O., Macedo, J., and Madeira, H. (2008). PorTexTO: sistema de anotação/extracção de expressões temporais. In Mota, C. and Santos, D., editors, *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*. Linguateca.

[Craveiro et al., 2009] Craveiro, O., Macedo, J., and Madeira, H. (2009). Use of co-occurrences for temporal expressions annotation. In *Proceedings of the 16th International Symposium on String Processing and Information Retrieval*, pages 156–164, Saariselka, Finland. Springer.

[Craveiro et al., 2010] Craveiro, O., Macedo, J., and Madeira, H. (2010). Leveraging temporal expressions for segmented-based information retrieval. In *ISDA'2010*, pages 754–759.

[Craveiro et al., 2012] Craveiro, O., Macedo, J., and Madeira, H. (2012). It is the time for Portuguese texts! In *PROPOR'2012*, pages 106–112.

[Craveiro et al., 2014a] Craveiro, O., Macedo, J., and Madeira, H. (2014a). Query expansion with temporal segmented texts. In *Advances in Information Retrieval - 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings*, pages 612–617.

[Craveiro et al., 2014b] Craveiro, O., Macedo, J., and Madeira, H. (2014b). Words temporality for improving query expansion. In *Computational Processing of the Portuguese Language - 11th International Conference, PROPOR 2014, Sao Carlos/SP, Brazil, October 6-8, 2014. Proceedings*, pages 262–267.

[Craveiro et al., 2015] Craveiro, O., Macedo, J., and Madeira, H. (2015). Temporal analysis of CHAVE collection. In *String Processing and Information Retrieval - 22nd International Symposium, SPIRE 2015, London, UK, September 1-4, 2015, Proceedings*, pages 67–74.

[Dai and Davison, 2010] Dai, N. and Davison, B. D. (2010). Freshness matters: in flowers, food, and web authority. In *SIGIR'10: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 114–121.

[Dakka et al., 2008] Dakka, W., Gravano, L., and Ipeirotis, P. G. (2008). Answering general time sensitive queries. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1437–1438, New York, NY, USA. ACM.

[de Jong et al., 2005] de Jong, F., Rode, H., and Hiemstra, D. (2005). Temporal language models for the disclosure of historical text. In *Humanities, computers and cultural heritage: Proceedings of the 16th International Conference of the Association for History and Computing (AHC'2005)*, pages 161–168, Amsterdam, The Netherlands. Royal Netherlands Academy of Arts and Sciences.

[Diaz and Jones, 2004] Diaz, F. and Jones, R. (2004). Using temporal profiles of queries for precision prediction. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'04, pages 18–24, New York, NY, USA. ACM.

[Doszkocs, 1978] Doszkocs, T. E. (1978). An associative interactive dictionary for online bibliographic searching. In *Jerusalem Conference on Information Technology*, pages 489–492.

[Efron et al., 2012] Efron, M., Organisciak, P., and Fenlon, K. (2012). Improving retrieval of short texts through document expansion. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 911–920. ACM.

[Ferro, 2001] Ferro, L. (2001). Tides: Instruction manual for the annotation of temporal expressions. Technical Report MTR 01W0000046V01, The MITRE Corporation.

[Filatova and Hovy, 2001] Filatova, E. and Hovy, E. (2001). Assigning time-stamps to event-clauses. In *Proceedings of the workshop on Temporal and spatial information processing*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

[Gupta and Berberich, 2014] Gupta, D. and Berberich, K. (2014). Identifying time intervals of interest to queries. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, CIKM'14, pages 1835–1838, New York, NY, USA. ACM.

[Gupta and Berberich, 2015] Gupta, D. and Berberich, K. (2015). Temporal query classification at different granularities. In Iliopoulos, C., Puglisi, S., and Yilmaz, E., editors, *String Processing and Information Retrieval*, volume 9309 of *Lecture Notes in Computer Science*, pages 156–164. Springer.

[Hagège and Tannier, 2008] Hagège, C. and Tannier, X. (2008). XTM: A robust temporal text processor. In *Computational Linguistics and Intelligent Text Processing, proceedings of 9th*

International Conference CICLing 2008, pages 231–240, Haifa, Israel. Springer Berlin / Heidelberg.

[Hagège et al., 2008a] Hagège, C., Baptista, J., and Mamede, N. (2008a). Apêndice B: Proposta de anotação e normalização de expressões temporais da categoria TEMPO para o Segundo HAREM. In Mota, C. and Santos, D., editors, *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*. Linguateca.

[Hagège et al., 2008b] Hagège, C., Baptista, J., and Mamede, N. (2008b). Reconhecimento de entidades mencionadas com o XIP: Uma colaboração entre a Xerox e o L2F do INESC-ID Lisboa. In Mota, C. and Santos, D., editors, *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*. Linguateca.

[Hagège et al., 2010] Hagège, C., Baptista, J., and Mamede, N. J. (2010). Caracterização e processamento de expressões temporais em Português. *Linguamática*, 2(1):63–76.

[Hearst, 1994] Hearst, M. A. (1994). Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 9–16, Morristown, NJ, USA. Association for Computational Linguistics.

[Hearst, 1997] Hearst, M. A. (1997). Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, pages 33–64.

[Jatowt et al., 2005] Jatowt, A., Kawai, Y., and Tanaka, K. (2005). Temporal ranking of search engine results. In Ngu, A., Kitsuregawa, M., Neuhold, E., Chung, J.-Y., and Sheng, Q., editors, *WISE'05 Proceedings of the 6th International Conference on Web Information Systems Engineering*, volume 3806 of *Lecture Notes in Computer Science*, pages 43–52. Springer Berlin Heidelberg.

[Jean-Louis et al., 2010]Jean-Louis, L., Besançon, R., and Ferret, O. (2010). Using temporal cues for segmenting texts into events. In *Proceedings of the 7th international conference on Advances in natural language processing*, IceTAL'10, pages 150–161, Berlin, Heidelberg. Springer-Verlag.

[Ji and Zha, 2003] Ji, X. and Zha, H. (2003). Domain-independent text segmentation using anisotropic diffusion and dynamic programming. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR'03, pages 322–329, New York, NY, USA. ACM.

[Jones and Diaz, 2007] Jones, R. and Diaz, F. (2007). Temporal profiles of queries. *ACM Trans. Inf. Syst.*, 25(3):14.

[Kalczynski and Chou, 2005] Kalczynski, P. J. and Chou, A. (2005). Temporal document retrieval model for business news archives. *Information Processing and Management*, 41(3):635–650.

[Kanhabua and Nørvåg, 2008] Kanhabua, N. and Nørvåg, K. (2008). Improving temporal language models for determining time of non-timestamped documents. In *ECDL '08: Proceedings of the 12th European conference on Research and Advanced Technology for Digital Libraries*, pages 358–370, Berlin, Heidelberg. Springer-Verlag.

[Kanhabua and Nørvåg, 2012] Kanhabua, N. and Nørvåg, K. (2012). Learning to rank search results for time-sensitive queries. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM'12, pages 2463–2466, New York, NY, USA. ACM.

[Keikha et al., 2011a] Keikha, M., Gerani, S., and Crestani, F. (2011a). TEMPER: A temporal relevance feedback method. In Clough, P., Foley, C., Gurrin, C., Jones, G., Kraaij, W., Lee, H., and Mudoch, V., editors, *Advances in Information Retrieval*, volume 6611 of *Lecture Notes in Computer Science*, pages 436–447. Springer Berlin Heidelberg.

[Keikha et al., 2011b] Keikha, M., Gerani, S., and Crestani, F. (2011b). Time-based relevance models. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR'11, pages 1087–1088, New York, NY, USA. ACM.

[Labadié and Prince, 2008] Labadié, A. and Prince, V. (2008). Lexical and semantic methods in inner text topic segmentation: A comparison between c99 and transeg. In *NLDB '08: Proceedings of the 13th international conference on Natural Language and Information Systems*, pages 347–349, Berlin, Heidelberg. Springer-Verlag.

[Lan et al., 2014] Lan, C., Zhang, Y., Xing, C., and Li, C. (2014). Continuous temporal top-k query over versioned documents. In Li, F., Li, G., Hwang, S.-w., Yao, B., and Zhang, Z., editors, *Web-Age Information Management*, volume 8485 of *Lecture Notes in Computer Science*, pages 494–497. Springer International Publishing.

[Lavrenko and Croft, 2001] Lavrenko, V. and Croft, W. B. (2001). Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'01, pages 120–127, New York, NY, USA. ACM.

[Li and Croft, 2003] Li, X. and Croft, W. B. (2003). Time-based language models. In *CIKM '03: Proceedings of the 12th international conference on Information and knowledge management*, pages 469–475, New York, NY, USA. ACM.

[Lin et al., 2014] Lin, S., Jin, P., Zhao, X., and Yue, L. (2014). Exploiting temporal information in web search. *Expert Syst. Appl.*, 41(2):331–341.

[Macdonald et al., 2012] Macdonald, C., McCreadie, R., Santos, R. L. T., and Ounis, I. (2012). From puppy to maturity: Experiences in developing Terrier. In *Proceedings of the SIGIR 2012 Workshop on Open Source Information Retrieval*, pages 60–63, Portland, OR, USA.

[Makkonen and Ahonen-Myka, 2003] Makkonen, J. and Ahonen-Myka, H. (2003). Utilizing temporal information in topic detection and tracking. In *ECDL'2003:7th European Conference on Digital Libraries*, pages 393–404.

[Mani, 2003] Mani, I. (2003). Recent developments in temporal information extraction. In *RANLP*, pages 45–60.

[Mani and Wilson, 2000] Mani, I. and Wilson, G. (2000). Robust temporal processing of news. In *ACL '00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 69–76, Morristown, NJ, USA. Association for Computational Linguistics.

[Manning et al., 2008] Manning, C. D., Raghavan, P., and Schtze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

[Martins and Silva, 2004] Martins, B. and Silva, M. J. (2004). A statistical study of the WPT-03 corpus. In *Advances in Natural Language Processing, 4th International Conference, EsTAL 2004, Alicante, Spain, October 20-22, 2004, Proceedings*, pages 384–394.

[McCreadie et al., 2012]McCreadie, R., Macdonald, C., and Ounis, I. (2012). MapReduce indexing strategies: Studying scalability and efficiency. *Inf. Process. Manage.*, 48(5):873–888.

[McCreadie et al., 2009]McCreadie, R., Macdonald, C., Ounis, I., Peng, J., and Santos, R. L. T. (2009). University of Glasgow at TREC 2009: Experiments with Terrier. In Voorhees, E. M. and Buckland, L. P., editors, *TREC*, volume Special Publication 500-278. National Institute of Standards and Technology (NIST).

[Metzler et al., 2009] Metzler, D., Jones, R., Peng, F., and Zhang, R. (2009). Improving search relevance for implicitly temporal queries. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 700–701, New York, NY, USA. ACM.

[Móia, 2001] Móia, T. (2001). Telling apart temporal locating adverbials and time-denoting expressions. In *Proceedings of the workshop on Temporal and spatial information processing*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

[Móia, 2006] Móia, T. (2006). Portuguese expressions of duration and its English counterparts. *Journal of Portuguese Linguistics*, 5:37–73.

[Middleton and Baeza-Yates, 2007] Middleton, C. and Baeza-Yates, R. (2007). A comparison of open source search engines. Technical report.

[Misra et al., 2011] Misra, H., Yvon, F., Cappé, O., and Jose, J. M. (2011). Text segmentation: A topic modeling perspective. *Inf. Process. Manage.*, 47(4):528–544.

[Misra et al., 2009] Misra, H., Yvon, F., Jose, J. M., and Cappe, O. (2009). Text segmentation via topic modeling: an analytical study. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1553–1556, New York, NY, USA. ACM.

[Mota and Santos, 2008] Mota, C. and Santos, D., editors (2008). *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas*. Linguateca.

[Mota et al., 2008a] Mota, C., Santos, D., Carvalho, P., Freitas, C., and Oliveira, H. G. (2008a). Apêndice H: Apresentação detalhada das colecções. In Mota, C. and Santos, D., editors, *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*. Linguateca.

[Mota et al., 2008b] Mota, C., Santos, D., Carvalho, P., Freitas, C., and Oliveira, H. G. (2008b). Apêndice I: Resumo de resultados do Segundo HAREM. In Mota, C. and Santos, D., editors, *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*. Linguateca.

[Nunes et al., 2007] Nunes, S., Ribeiro, C., and David, G. (2007). Using neighbors to date Web documents. In *Proceedings of the 9th annual ACM international workshop on Web information and data management*, WIDM'07, pages 129–136, New York, NY, USA. ACM.

[Oliveira et al., 2008] Oliveira, H. G., Mota, C., Freitas, C., Santos, D., and Carvalho, P. (2008). Avaliação à medida no Segundo HAREM. In Mota, C. and Santos, D., editors, *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*, chapter 5. Linguateca.

[Ounis et al., 2005] Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., and Johnson, D. (2005). Terrier information retrieval platform. In *Proceedings of the 27th European*

Conference on IR Research (ECIR 2005), volume 3408 of *Lecture Notes in Computer Science*, pages 517–519. Springer.

[Ounis et al., 2006] Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., and Lioma, C. (2006). Terrier: A high performance and scalable information retrieval platform. In *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*, Seattle, Washington, USA.

[Ounis et al., 2007] Ounis, I., Lioma, C., Macdonald, C., and Plachouras, V. (2007). Research directions in Terrier: a search engine for advanced retrieval on the Web. *Novatica/UPGRADE Special Issue on Web Information Access, Ricardo Baeza-Yates et al. (Eds), Invited Paper*, 8(1):49–56.

[Pan et al., 2013] Pan, S., Zhou, M. X., Song, Y., Qian, W., Wang, F., and Liu, S. (2013). Optimizing temporal topic segmentation for intelligent text visualization. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, IUI'13, pages 339–350, New York, NY, USA. ACM.

[Pant et al., 2004] Pant, G., Srinivasan, P., and Menczer, F. (2004). Crawling the Web. In *In Web Dynamics: Adapting to Change in Content, Size, Topology and Use. Edited by M. Levene and A. Poulovassilis*, pages 153–178. Springer-Verlag.

[Pereira et al., 2014] Pereira, P., Macedo, J., Craveiro, O., and Madeira, H. (2014). Time-aware focused Web crawling. In *Advances in Information Retrieval - 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings*, pages 534–539.

[Pereira, 2013] Pereira, P. V. N. (2013). Descarga temporal de páginas Web. Master's thesis, Universidade do Minho, Departamento de Engenharia Informática.

[Perona and Malik, 1990] Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(7):629–639.

[Pevzner and Hearst, 2002] Pevzner, L. and Hearst, M. A. (2002). A critique and improvement of an evaluation metric for text segmentation. *Comput. Linguist.*, 28:19–36.

[Ponte and Croft, 1997] Ponte, J. M. and Croft, W. B. (1997). Text segmentation by topic. In *ECDL '97: Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries*, pages 113–125, London, UK. Springer-Verlag.

[Pustejovsky et al., 2005] Pustejovsky, J., Ingria, B., Sauri, R., Castano, J., Littman, J., Gaizauskas, R., Setzera, A., Katz, G., and Mani, I. (2005). *The Language of Time: A Reader*, chapter The Specification Language TimeML. Oxford University Press.

[Radinsky et al., 2011] Radinsky, K., Agichtein, E., Gabrilovich, E., and Markovitch, S. (2011). A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, WWW'11, pages 337–346, New York, NY, USA. ACM.

[Robertson, 1977] Robertson, S. E. (1977). The probability ranking principle in IR. *Journal of Documentation*, 33(4):294–304.

[Robertson, 1990] Robertson, S. E. (1990). On term selection for query expansion. *J. Doc.*, 46(4):359–364.

[Robertson and Jones, 1976] Robertson, S. E. and Jones, K. S. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146.

[Rocchio, 1971]Rocchio, J. (1971). Relevance feedback in information retrieval. In Salton, G., editor, *The SMART Retrieval System—Experiment in Automatic Document Processing*, pages 313–323. Prentice-Hall, New Jersey.

[Rode and Hiemstra, 2006] Rode, H. and Hiemstra, D. (2006). Using query profiles for clarification. In *Proceedings of the 28th European Conference on Information Retrieval ECIR*, pages 205–216.

[Salton and Buckley, 1988] Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523.

[Salton et al., 1985] Salton, G., Fox, E. A., and Voorhees, E. (1985). Advanced feedback methods in information retrieval. *Journal of the American Society for Information Science*, 36(3):200–210.

[Salton et al., 1975] Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.

[Santos and Cardoso, 2007] Santos, D. and Cardoso, N., editors (2007). *Reconhecimento de entidades mencionadas em Português: Documentação e actas do HAREM, a primeira avaliação conjunta na área*. Linguateca.

[Santos and Rocha, 2004] Santos, D. and Rocha, P. (2004). CHAVE: topics and questions on the Portuguese participation in CLEF. In Peters, C. and Borri, F., editors, *Cross Language*

Evaluation Forum: Working Notes for the CLEF 2004 Workshop (CLEF 2004), pages 639–648, Pisa, Italy. IST-CNR. Revised as Santos & Rocha (2005).

[Santos and Rocha, 2005] Santos, D. and Rocha, P. (2005). *The key to the first CLEF in Portuguese: Topics, questions and answers in CHAVE*, volume 3491 of *Lecture Notes in Computer Science*, pages 821–832. Springer, Berlin/Heidelberg. Revised version of Santos & Rocha (2004).

[Sato et al., 2003] Sato, N., Uehara, M., and Sakai, Y. (2003). Temporal ranking for fresh information retrieval. In *Proceedings of the sixth international workshop on Information retrieval with Asian languages*, pages 116–123, Morristown, NJ, USA. Association for Computational Linguistics.

[Saurí et al., 2006] Saurí, R., Littman, J., Knippen, B., Gaizauskas, R., Setzer, A., and Pustejovsky, J. (2006). TimeML annotation guidelines, version 1.2.1. Technical report.

[Saurí and Pustejovsky, 2009] Saurí, R. and Pustejovsky, J. (2009). TimeML in a nutshell. electronic, http://www.timeml.org/tempeval2/tempeval2-trial/guidelines/introToTimeML-052809.pdf (retrieved September 20, 2014).

[Schilder and Habel, 2001] Schilder, F. and Habel, C. (2001). From temporal expressions to temporal information: Semantic tagging of news messages. In *Proceedings of ACL'01 workshop on temporal and spatial information processing*, pages 65–72, Toulouse, France.

[Setzer, 2001] Setzer, A. (2001). *Temporal Information in Newswire Articles: an Annotation Scheme and Corpus Study*. PhD thesis, University of Sheffield, Sheffield, UK.

[Shokouhi, 2011] Shokouhi, M. (2011). Detecting seasonal queries by time-series analysis. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR'11, pages 1171–1172, New York, NY, USA. ACM.

[Sparck Jones and Willett, 1997] Sparck Jones, K. and Willett, P., editors (1997). *Readings in information retrieval*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Strötgen and Gertz, 2013] Strötgen, J. and Gertz, M. (2013). Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298.

[Strötgen and Gertz, 2015] Strötgen, J. and Gertz, M. (2015). A baseline temporal tagger for all languages. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 541–547, Lisbon, Portugal. Association for Computational Linguistics.

[Vazov, 2001] Vazov, N. (2001). A system for extraction of temporal expressions from french texts based on syntactic and semantic constraints. In *Proceedings of ACL-2001: Workshop on Temporal and Spatial Information Processing*.

[Verhagen and Pustejovsky, 2008] Verhagen, M. and Pustejovsky, J. (2008). Temporal processing with the TARSQI toolkit. In *COLING '08: 22nd International Conference on on Computational Linguistics: Demonstration Papers*, pages 189–192, Morristown, NJ, USA. Association for Computational Linguistics.

[Whiting et al., 2011] Whiting, S., Moshfeghi, Y., and Jose, J. M. (2011). Exploring term temporality for pseudo-relevance feedback. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR'11, pages 1245–1246, New York, NY, USA. ACM.

[Wong et al., 2008] Wong, W., Luk, R., Leong, H., Ho, K., and Lee, D. (2008). Re-examining the effects of adding relevance information in a relevance feedback environment. *Information Processing & Management*, 44(3):1086 – 1116.

[Xu and Croft, 1996] Xu, J. and Croft, W. B. (1996). Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'96, pages 4–11, New York, NY, USA. ACM.

[Xu et al., 2009] Xu, Y., Jones, G. J., and Wang, B. (2009). Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'09, pages 59–66, New York, NY, USA. ACM.

[Yu et al., 2004] Yu, P. S., Li, X., and Liu, B. (2004). On the temporal dimension of search. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, WWW '04*, pages 448—449.

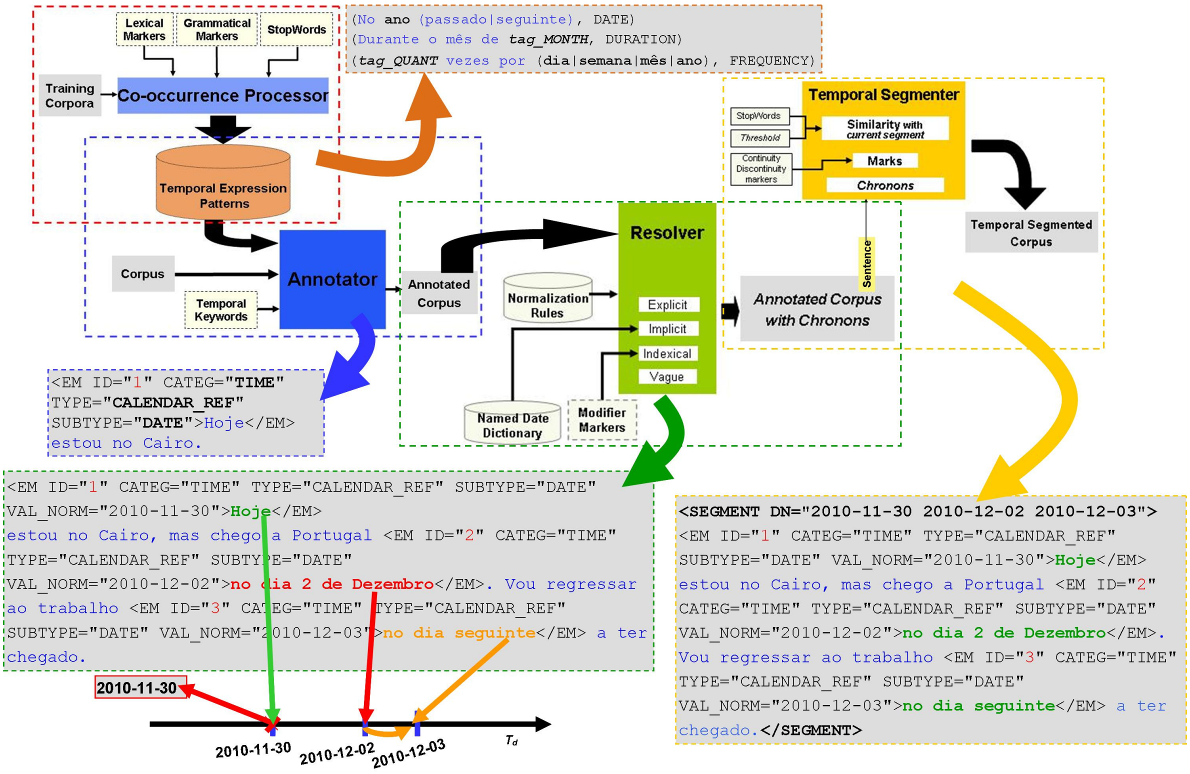
Appendix A

Example of the Temporal Processing in Portuguese Texts

In this appendix, we present all the steps used in the preprocessing of the temporal retrieval. First, the Annotator module annotates the temporal expressions found in the content of the original documents. Then, Resolver uses these annotations to map them into *chronons*. Finally, the temporal segmentation is carried out by the Segmenter module, taking into account the obtained *chronons*. Note that the preprocessing can only starts after the temporal expressions patterns are created.

Figure A.1 shows the tested system architecture with the following examples[35]: (1) "*Hoje estou no Cairo.*", and (2) *"Hoje estou no Cairo, mas chego a Portugal no dia 2 de Dezembro. Vou regressar ao trabalho no dia seguinte a ter chegado."*. The examples of temporal expression patterns[36] showed by this figure are the following:

(1) (No **ano** (passado|seguinte), DATE), (2) (Durante o mês de ***tag_MONTH*,** DURATION), and (3) (***tag_QUANT*** vezes por (**dia|semana|mês|ano**), FREQUENCY).



```
(No ano (passado|seguinte), DATE)
(Durante o mês de tag_MONTH, DURATION)
(tag_QUANT vezes por (dia|semana|mês|ano), FREQUENCY)
```

```
<EM ID="1" CATEG="TIME"
TYPE="CALENDAR_REF"
SUBTYPE="DATE">Hoje</EM>
estou no Cairo.
```

```
<EM ID="1" CATEG="TIME" TYPE="CALENDAR_REF" SUBTYPE="DATE"
VAL_NORM="2010-11-30">Hoje</EM>
estou no Cairo, mas chego a Portugal <EM ID="2" CATEG="TIME"
TYPE="CALENDAR_REF" SUBTYPE="DATE"
VAL_NORM="2010-12-02">no dia 2 de Dezembro</EM>. Vou regressar
ao trabalho <EM ID="3" CATEG="TIME" TYPE="CALENDAR_REF"
SUBTYPE="DATE" VAL_NORM="2010-12-03">no dia seguinte</EM> a ter
chegado.
```

```
<SEGMENT DN="2010-11-30 2010-12-02 2010-12-03">
<EM ID="1" CATEG="TIME" TYPE="CALENDAR_REF"
SUBTYPE="DATE" VAL_NORM="2010-11-30">Hoje</EM>
estou no Cairo, mas chego a Portugal <EM ID="2"
CATEG="TIME" TYPE="CALENDAR_REF" SUBTYPE="DATE"
VAL_NORM="2010-12-02">no dia 2 de Dezembro</EM>.
Vou regressar ao trabalho <EM ID="3" CATEG="TIME"
TYPE="CALENDAR_REF" SUBTYPE="DATE"
VAL_NORM="2010-12-03">no dia seguinte</EM> a ter
chegado.</SEGMENT>
```

Figure A.1: Testbed system architecture with some examples.

[35] English version: (1) *Today I am in Cairo* (2) *Today I am in Cairo, but I arrive in Portugal on 2nd of December. I return to work in the day after I arrive.*

[36] English version: (1) In the (last|following) year (2) During the month of tag_MONTH
(3) tag_QUANT times per (day|week|month|year)

We use two documents of the CHAVE collection, *PUBLICO-19940106-011* and *PUBLICO-19940107-015*, which were processed by our temporal system, to showcase the input and output of each of the tools introduced in Chapter 3. The tools are also displayed in Figure A.1.

The processing starts with the annotation of the temporal expressions found in original documents (see Figure A.2) performed by the Annotator Module. The output is the annotated documents displayed in Figure A.3.

```
<DOC>
<CATEGORY>Cultura</CATEGORY>
<DATE>19940106</DATE>
<DOCID>PUBLICO-19940106-011</DOCID>
<DOCNO>PUBLICO-19940106-011</DOCNO>
<TEXT>
<P> Antiguidades voltam à Grécia</P>
<P> A Grécia conseguiu que uma galeria de Arte, de Nova Iorque, lhe
restituísse cinquenta peças de antiguidades micénicas provenientes do
norte do Peloponeso, que tinham desaparecido há meio século. A ministra
grega da cultura, Mélina Mercouri, declarou estar «muito contente» com a
restituição, que «constitui uma vitória dos serviços arqueológicos
gregos». A galeria de arte aceitou devolver as peças depois de uma
delegação de arqueólogos gregos ter ido a Nova Iorque e ter provado que
as peças haviam sido retiradas da Grécia, de onde desapareceram.</P>
</TEXT>
</DOC>

<DOC>
<AUTHOR>PFER</AUTHOR>
<CATEGORY>Economia</CATEGORY>
<DATE>19940107</DATE>
<DOCID>PUBLICO-19940107-015</DOCID>
<DOCNO>PUBLICO-19940107-015</DOCNO>
<TEXT>
<P> Final previsto</P>
<P> Pela terceira sessão consecutiva, a Bolsa de Frankfurt fechou em
baixa, com o índice Dax-30 a quebra. Cotou-se nos 2220,22 pontos, menos
0,59 por cento. Operadores do mercado alemão afirmaram que se tratou de
mais uma correcção, isto porque as decisões do Bundesbank já eram
esperados, razão pela qual tiveram uma influência nula. O mercado deverá
agora começar a subir, porque ontem já terminou acima dos mínimos
apurados durante o dia.</P>
</TEXT>
</DOC>
```

Figure A.2: Documents in original format.

```
<DOC>
<CATEGORY>Cultura</CATEGORY>
<DATE>19940106</DATE>
<DOCID>PUBLICO-19940106-011</DOCID>
<DOCNO>PUBLICO-19940106-011</DOCNO>
<TEXT>
<P> Antiguidades voltam à Grécia</P>
<P> A Grécia conseguiu que uma galeria de Arte, de Nova Iorque, lhe
restituísse cinquenta peças de antiguidades micénicas provenientes do
norte do Peloponeso, que tinham desaparecido <EM ID="429685"
CATEG="TEMPO" TIPO="TEMPO_CALEND" SUBTIPO="DATA">há meio século</EM>. A
ministra grega da cultura, Mélina Mercouri, declarou estar «muito
contente» com a restituição, que «constitui uma vitória dos serviços
arqueológicos gregos». A galeria de arte aceitou devolver as peças depois
de uma delegação de arqueólogos gregos ter ido a Nova Iorque e ter
provado que as peças haviam sido retiradas da Grécia, de onde
desapareceram <EM ID="429686" CATEG="TEMPO" TIPO="TEMPO_CALEND"
SUBTIPO="DATA">nos anos 30</EM>.</P>
</TEXT>
</DOC>

<DOC>
<AUTHOR>PFER</AUTHOR>
<CATEGORY>Economia</CATEGORY>
<DATE>19940107</DATE>
<DOCID>PUBLICO-19940107-015</DOCID>
<DOCNO>PUBLICO-19940107-015</DOCNO>
<TEXT>
<P> Final previsto</P>
<P> Pela terceira sessão consecutiva, a Bolsa de Frankfurt fechou em
baixa, com o índice Dax-30 a quebra. Cotou-se nos 2220,22 pontos, menos
0,59 por cento. Operadores do mercado alemão afirmaram que se tratou de
mais uma correcção, isto porque as decisões do Bundesbank já eram
esperados, razão pela qual tiveram uma influência nula. O mercado deverá
agora começar a subir, porque <EM ID="430619" CATEG="TEMPO"
TIPO="TEMPO_CALEND" SUBTIPO="DATA">ontem</EM> já terminou acima dos
mínimos apurados <EM ID="430618" CATEG="TEMPO" TIPO="DURACAO">durante o
dia</EM>.</P>
</TEXT>
</DOC>
```

Figure A.3: Annotated documents.

Then, the annotated documents (see Figure A.3) are processed by the Resolver module, mapping the annotated temporal expressions into *chronons,* whenever possible, and adding the normalized value to their annotation. Note that the expressions[37] *"há meio século"* and *"durante o dia"* are not resolved, since this module still unable to normalize all of temporal expression, as explained in Chapter 3. Figure A.4 shows this module output.

[37] English version: *"an half century ago"* and *"during the day"*

```
<DOC>
<CATEGORY>Cultura</CATEGORY>
<DATE>19940106</DATE>
<DOCID>PUBLICO-19940106-011</DOCID>
<DOCNO>PUBLICO-19940106-011</DOCNO>
<TEXT>
<P> Antiguidades voltam à Grécia</P>
<P> A Grécia conseguiu que uma galeria de Arte, de Nova Iorque, lhe
restituísse cinquenta peças de antiguidades micénicas provenientes do
norte do Peloponeso, que tinham desaparecido <EM ID="429685"
CATEG="TEMPO" TIPO="TEMPO_CALEND" SUBTIPO="DATA">há meio século</EM>. A
ministra grega da cultura, Mélina Mercouri, declarou estar «muito
contente» com a restituição, que «constitui uma vitória dos serviços
arqueológicos gregos». A galeria de arte aceitou devolver as peças depois
de uma delegação de arqueólogos gregos ter ido a Nova Iorque e ter
provado que as peças haviam sido retiradas da Grécia, de onde
desapareceram <EM ID="429686" CATEG="TEMPO" TIPO="TEMPO_CALEND"
SUBTIPO="DATA" VAL_NORM="193X-XX-XX">nos anos 30</EM>.</P>
</TEXT>
</DOC>

<DOC>
<AUTHOR>PFER</AUTHOR>
<CATEGORY>Economia</CATEGORY>
<DATE>19940107</DATE>
<DOCID>PUBLICO-19940107-015</DOCID>
<DOCNO>PUBLICO-19940107-015</DOCNO>
<TEXT>
<P> Final previsto</P>
<P> Pela terceira sessão consecutiva, a Bolsa de Frankfurt fechou em
baixa, com o índice Dax-30 a quebra. Cotou-se nos 2220,22 pontos, menos
0,59 por cento. Operadores do mercado alemão afirmaram que se tratou de
mais uma correcção, isto porque as decisões do Bundesbank já eram
esperados, razão pela qual tiveram uma influência nula. O mercado deverá
agora começar a subir, porque <EM ID="430619" CATEG="TEMPO"
TIPO="TEMPO_CALEND" SUBTIPO="DATA" VAL_NORM="1994-01-06">ontem</EM> já
terminou acima dos mínimos apurados <EM ID="430618" CATEG="TEMPO"
TIPO="DURACAO">durante o dia</EM>.</P>
</TEXT>
</DOC>
```

Figure A.4: Annotated documents with *chronons*.

Finally, the Segmenter module comes in. The documents coming from the Resolver module output (see Figure A.4) are segmented taking into account the temporal discontinues found in their content, as explained in Chapter 3. Figure A.5 shows the temporal segmented documents.

```
<DOC>
<CATEGORY>Cultura</CATEGORY>
<DATE>19940106</DATE>
<DOCID>PUBLICO-19940106-011</DOCID>
<DOCNO>PUBLICO-19940106-011</DOCNO>
<TEXT>
<P><SEGMENT> Antiguidades voltam à Grécia</SEGMENT></P>
<P><SEGMENT> A Grécia conseguiu que uma galeria de Arte, de Nova Iorque,
lhe restituísse cinquenta peças de antiguidades micénicas provenientes do
norte do Peloponeso, que tinham desaparecido <EM ID="429685"
CATEG="TEMPO" TIPO="TEMPO_CALEND" SUBTIPO="DATA">há meio século</EM>.
</SEGMENT>
<SEGMENT DN="193X-XX-XX">A ministra grega da cultura, Mélina Mercouri,
declarou estar «muito contente» com a restituição, que «constitui uma
vitória dos serviços arqueológicos gregos». A galeria de arte aceitou
devolver as peças depois de uma delegação de arqueólogos gregos ter ido a
Nova Iorque e ter provado que as peças haviam sido retiradas da Grécia,
de onde desapareceram <EM ID="429686" CATEG="TEMPO" TIPO="TEMPO_CALEND"
SUBTIPO="DATA" VAL_NORM="193X-XX-XX">nos anos 30</EM>.</SEGMENT></P>
</TEXT>
</DOC>

<DOC>
AUTHOR>PFER</AUTHOR>
<CATEGORY>Economia</CATEGORY>
<DATE>19940107</DATE>
<DOCID>PUBLICO-19940107-015</DOCID>
<DOCNO>PUBLICO-19940107-015</DOCNO>
<TEXT>
<P><SEGMENT> Final previsto</SEGMENT></P>
<P><SEGMENT> Pela terceira sessão consecutiva, a Bolsa de Frankfurt
fechou em baixa, com o índice Dax-30 a quebra.</SEGMENT>
<SEGMENT>Cotou-se nos 2220,22 pontos, menos 0,59 por cento.</SEGMENT>
<SEGMENT DN="1994-01-06">Operadores do mercado alemão afirmaram que se
tratou de mais uma correcção, isto porque as decisões do Bundesbank já
eram esperados, razão pela qual tiveram uma influência nula. O mercado
deverá agora começar a subir, porque <EM ID="430619" CATEG="TEMPO"
TIPO="TEMPO_CALEND" SUBTIPO="DATA" VAL_NORM="1994-01-06">ontem</EM> já
terminou acima dos mínimos apurados <EM ID="430618" CATEG="TEMPO"
TIPO="DURACAO">durante o dia</EM>.</SEGMENT></P>
</TEXT>
</DOC>
```

Figure A.5: Temporal segmented documents.

Appendix B

Segmenter Module Lists

e
ela
elas
ele
eles
essa
essas
esse
esses
esta
estas
este
estes
isso
isto
nessa
nessas
nesse
nesses
nesta
nestas
neste
nestes
também

Table B.1: Continuity marks list of the Segmenter module.

afinal
agora
ainda
amanhã
amiúde
anteontem
antes
antigamente
após
aqui
breve
brevemente
cedo
comumente
concomitantemente
dantes
depois
diariamente
doravante
em seguida
enfim
então
entrementes
finalmente
hoje
imediatamente
já
jamais
logo
no final
nunca
ontem
ora
outrora
presentemente
primeiramente
primeiro
quando
raramente
sempre
simultaneamente
sucessivamente
tarde

Table B.2: Discontinuity marks list of the Segmenter module**.**

a
à
acima
ainda
algo
alguém
algum
alguma
algumas
alguns
ambas
ambos
ao
ao
aos
apesar
após
aquela
aquele
aqueles
aquilo
as
às
assim
até
cada
caso
cento
cerca
certa
certo
certos
com
como
conforme
contanto
contra
contudo
cujo
cujos
da
daquela
daquelas
daquele
daqueles
daquilo
das
de
dela
delas
dele
deles
demais
desde
dessa
dessas
desse
desses
desta
destas
deste
destes
devido
diante
disso
dissonante
diversas
diversos
do
dólares
dos
dourada
e
ela
elas
ele
eles
em
embora
enquanto
entanto
então
entre
entretanto
essa
essas
esse
esses
esta
estas
este
estes
excepto
exceto
há
havendo
havido
isso
isto
la
las
lhe
lhes
libre
lo
los
mais
mas
me
menos
mesma
mesmas
mesmo
mesmos
meu
meus
mim
minha
minhas
modo
monte
muita
muitas
muito
muitos
na
nada
não
nas
né
nem
nenhum
nenhuma
nessa
ninguém
no
nos
nós
nossa
nossas
nosso
nossos
num
numa
o
obstante
onde
os
ou
outra
outras
outro
outros
para
pela
pelas
pelo
pelos
per
perante
pois
por
porém
porque
porquê
portanto
posto
poucas
pouco
poucos
pousada
pra
prá
própria
próprias
próprio
próprios
quais
qual
qualquer
quando
quantas
quanto
quantos
quase
que
quê
quem
quer
salvo
se
segundo
sem
sempre
senão
sendo
sendo
sentido
ser
serpente
seu
seus
si
sido
sob
sobre
soda
sua
suas
tais
tal
tanta
tantas
tanto
tantos
te
tem
tendo
teu
teus
ti
tido
toda
todas
todo
todos
tudo
um
uma
umas
uns
várias
vários
vez
voce
você
voces
vocês
vos
vos
vós
vossa
vossas

Table B.3: Stopwords list of the Segmenter module.

Appendix C

HAREM and CHAVE Collection Documents

```
<DOC DOCID="hub-28874">
<P>H5N1: Mais de 32 mil mortos se pandemia atingisse Portugal</P>
<P>Mais de 32 mil pessoas poderiam morrer se uma pandemia de gripe humana de
origem aviária atingisse Portugal, segundo cenários elaborados este ano por
peritos do Instituto Nacional de Saúde Ricardo Jorge.</P>
<P>Na quinta-feira, a Organização Mundial de Saúde confirmou que um
paquistanês vítima da gripe das aves tinha contraído o vírus H5N1 de outro
humano, apesar de os peritos afastarem ainda qualquer risco de contaminação
generalizado.</P>
<P>Em Portugal, o Instituto Nacional de Saúde elaborou cenários de uma
eventual pandemia de gripe humana de origem em aves, que poderá ou não ser
desencadeada pelo H5N1, a estirpe do vírus mais mortal até agora
conhecida.</P>
<P>Os cenários tiveram em conta a utilização do Oseltamivir (o anti-viral
tido como o mais eficaz contra uma eventual pandemia com origem da gripe das
aves), mas não consideraram outras medidas de saúde pública, apesar de os
autores admitirem que estas «terão um efeito principal, embora não
exclusivo, na diminuição da incidência da doença e, portanto, nas taxas de
ataque».</P>
<P>Os autores dos cenários basearam o seu cálculo em três taxas de ataque
(30, 35 e 40 por cento da população), admitindo que a pandemia evoluiria em
duas ondas.</P>
<P>Os peritos consideram provável que, com uma taxa de ataque de 30 por
cento, existiriam 3.106.835 casos de gripe, 3.624.641 numa taxa de ataque de
35 por cento e 4.142.447 perante a mais severa taxa de ataque (40 por
cento). </P>
</DOC>
```

Figure C.1: Document *hub-28874* from the Second HAREM collection.

```
<DOC>
<DOCNO>PUBLICO-19940127-135</DOCNO>
<DOCID>PUBLICO-19940127-135</DOCID>
<DATE>19940127</DATE>
<CATEGORY>Desporto</CATEGORY>
<AUTHOR>JMF</AUTHOR>
<TEXT>
BMW 325 tds
Você pediu diesel?
Ai, ai: por vezes há automóveis de que nos custa separar, quando os
devolvemos às marcas, depois de os testarmos. Este BMW 325 tds foi um desses
casos, talvez porque é raro guiar um modelo que possa aspirar a um título
mundial. Ora ele é, se calhar, o melhor carro com motor diesel do mundo.
Mas, ai: tudo isto custa muito, muito dinheiro -- quase dez mil contos.
Os BMW da nove série 3 caíram no goto dos portugueses. Num mercado como o
nosso, onde o factor preço é importante, o mais jovem membro da prestigiosa
família bávara conseguiu a notável «performance» de ser campeão de vendas na
sua categoria, em 1993. Em concorrência com modelos tão difundidos com o
Renault 21, o Peugeot 405, o VW Passat ou o Nissan Primera, os BMW da série
3 bateram-nos a todos, o que não deixa de ser notável quando estes modelos,
para níveis semelhantes de equipamento e potência, custam cerca de mil
contos mais que os seus concorrentes directos.
Para esta carreira de sucesso contribuíram vários factores, nomeadamente a
existência de um modelo de «entrada» na gama relativamente acessível -- o
316i --, uma estética que faz a unanimidade, casando de forma harmoniosa
classe e originalidade com um toque de agressividade desportiva, uma elevada
qualidade de construção e uma imagem de prestígio.
Da várias vezes que guiámos modelos da nova série 3 pudemos apreciar uma
meia dúzia de características que ajudam a compreender este sucesso. Entre
elas, destacamos uns interiores bem concebidos, um posto de condução
```

racionalmente desenhado e uma grande facilidade de condução, para o que contribui um equilíbrio no comportamento que faz esquecer o estarmos perante um veículo de tracção traseira. As caixas de velocidades firmes, mas muito fáceis de engrenar, a direcção, assistida, precisa e directa, uma suspensão que consegue uma boa filtragem das irregularidades do piso, sem por isso ser demasiado mole -- tudo são qualidades que tínhamos encontrado ao longo da gama.

Como contraponto, nunca gostámos do espaço deixado livre para os passageiros que seguem no banco de trás, obrigados a seguir numa posição pouco confortável devido ao desenho dos bancos da frente, cujas costas se prolongam até ao chão; o espaço a nível dos ombros na parte traseira, que também não é generoso; e a bagageira, que está longe de ser a mais espaçosa da categoria. Notámos igualmente uma tendência quase geral para consumos elevados, tendência essa «agravada» pelas características da gama, que convidam geralmente os condutores a imprimirem ritmos mais vivos, que logo induzem a um maior gasto de combustível.

Ora foi precisamente neste último capítulo que começámos por notar o valor da nova proposta da marca, um série 3 equipado com um motor diesel de seis cilindros, 2,5 litros, turbo comprimido e capaz de debitar 143 cavalos.

Concebido inicialmente para os BMW série 5, este propulsor é uma pequena jóia da tecnologia, tendo os técnicos alemães obtido um excelente rendimento através da utilização de um turbo moderno. Montado agora no mais pequeno 325, este motor consegue o pequeno prodígio de fazer deste BMW o diesel mais rápido do mundo. Mais de 210 km/h de velocidade máxima, pouco mais de 10 segundos para ir dos 0 aos 100 km/h e recuperações de velocidade tão vigorosas que conseguem mesmo ultrapassar as dos desportivo 325i.

Tudo isto faz deste diesel uma viatura de vocação desportiva. Surpreendente, numa palavra. Basta sentarmo-nos ao volante para sentirmos todo o vigor deste modelo, uma energia que faz esquecer por completo as referências clássicas, que desvalorizavam as motorizações diesel.

A insonorização é perfeita, a facilidade em subir de regime enorme, o nervosismo das reacções estimulante, a adaptação do motor à caixa de velocidades capaz de proporcionar todos os prazeres da condução desportiva. Ao lado de um 320i, de 150 cavalos, este 325 tds nunca fica a perder.

Este conjunto de características fazem deste modelo a referência que há que passar a ter em linha de conta, quando quisermos avaliar viaturas familiares equipadas com motor diesel. Mesmo os mais rápidos Mercedes, antes reis e senhores com os seus motores turbodiesel de três litros, têm agora de se inclinar perante esta pequena jóia de Munique.

Até porque, a todas estas qualidades, este propulsor ainda junta uma outra: uma enorme sobriedade. Mesmo em condução desportiva raramente ultrapassa a barreira dos dez litros aos cem (e são dez litros de gasóleo...), sendo possível fazer médias urbanas na casa dos oito litros. É uma verdadeiro apetite de passarinho.

Só é pena que, devido à sua elevada cilindrada, este modelo seja vendido a quase dez mil contos, devido à sobrecarga do Imposto Automóvel. São mais dois milhões de escudos que o 320i de prestações semelhantes, pelo que será necessário fazer muitos e muitos quilómetros (pelas nossas contas, entre 150 e 200 mil quilómetros) antes que o que se ganha nos reabastecimentos de combustíveis compense o que se paga a mais no «stand». Ai, ai...

</TEXT>
</DOC>

Figure C.2: Document *PUBLICO-19940127-135* of the CHAVE collection.

Appendix D

CHAVE Topics

	Documents Judged			Documents Judged			Documents Judged	
Topic	**NonRelevant**	**Relevant**	**Topic**	**NonRelevant**	**Relevant**	**Topic**	**NonRelevant**	**Relevant**
C201	482	14	**C251**	266	71	**C301**	258	28
C202	161	9	**C252**	542	44	**C302**	369	80
C203	332	37	**C253**	417	164	**C303**	389	50
C204	539	7	**C254**	181	129	**C304**	229	58
C205	167	2	**C255**	360	6	**C305**	347	63
C206	198	2	**C256**	311	12	**C306**	180	57
C207	576	12	**C257**	368	57	**C307**	586	33
C208	479	5	**C258**	574	2	**C308**	115	60
C209	633	2	**C259**	276	17	**C309**	371	43
C210	374	5	**C260**	236	89	**C310**	186	131
C211	215	28	**C261**	586	23	**C311**	178	181
C212	599	10	**C262**	697	33	**C312**	518	26
C213	226	56	**C263**	513	64	**C313**	307	215
C214	596	8	**C264**	231	43	**C314**	361	26
C215	349	1	**C265**	227	26	**C315**	435	82
C216	573	0	**C266**	300	87	**C316**	388	266
C217	577	7	**C267**	311	60	**C317**	360	63
C218	107	21	**C268**	372	7	**C318**	448	11
C219	654	2	**C269**	427	51	**C319**	449	37
C220	602	0	**C270**	199	41	**C320**	505	29
C221	290	6	**C271**	343	40	**C321**	186	40
C222	580	3	**C272**	308	17	**C322**	347	35
C223	386	1	**C273**	242	74	**C323**	608	56
C224	547	2	**C274**	388	6	**C324**	380	110
C225	436	1	**C275**	304	57	**C325**	575	68
C226	520	4	**C276**	347	39	**C326**	521	5
C227	476	0	**C277**	423	20	**C327**	444	4
C228	463	51	**C278**	486	19	**C328**	212	63
C229	316	189	**C279**	466	23	**C329**	289	12
C230	134	13	**C280**	509	34	**C330**	185	57
C231	828	6	**C281**	131	67	**C331**	208	41
C232	636	29	**C282**	366	86	**C332**	544	6
C233	250	16	**C283**	264	44	**C333**	309	29
C234	556	2	**C284**	282	40	**C334**	256	2
C235	452	1	**C285**	379	59	**C335**	279	36
C236	241	3	**C286**	430	239	**C336**	510	30
C237	384	3	**C287**	372	109	**C337**	161	94
C238	239	1	**C288**	441	8	**C338**	392	11
C239	429	27	**C289**	278	9	**C339**	90	81
C240	557	0	**C290**	411	66	**C340**	378	13
C241	939	58	**C291**	294	91	**C341**	492	14
C242	401	4	**C292**	430	8	**C342**	286	35
C243	382	1	**C293**	147	98	**C343**	262	70
C244	388	1	**C294**	235	33	**C344**	599	38
C245	290	3	**C295**	229	190	**C345**	384	14
C246	306	1	**C296**	111	49	**C346**	402	25
C247	523	8	**C297**	433	41	**C347**	396	11
C248	370	10	**C298**	281	76	**C348**	455	8
C249	345	4	**C299**	359	179	**C349**	197	40
C250	530	2	**C300**	552	57	**C350**	151	90
(1)	21,633	678	**(2)**	17,635	2,904	**(3)**	17,477	2,677

Table D.1: Number of documents judged for each CHAVE topic.

Group	Topic	TopicElement	TemporalExpression	NormalizedDate	Timeline
1	C206	PT-desc	em 1995	1995-XX-XX	Y
1	C209	PT-desc	de 1995	1995-XX-XX	Y
1	C209	PT-narr	em 1995	1995-XX-XX	Y
1	C210	PT-desc	de 1995	1995-XX-XX	Y
1	C213	PT-desc	em 1995	1995-XX-XX	Y
1	C213	PT-narr	em 1995	1995-XX-XX	Y
1	C215	PT-desc	em 1995	1995-XX-XX	Y
1	C221	PT-title	de 2002	2002-XX-XX	Y
1	C221	PT-desc	de 2002	2002-XX-XX	Y
1	C222	PT-desc	de 1995	1995-XX-XX	Y
1	C222	PT-narr	em Maio de 1995	1995-05-XX	M
1	C223	PT-narr	em Maio de 1995	1995-05-XX	M
1	C225	PT-desc	em 1995	1995-XX-XX	Y
1	C225	PT-narr	em 1995	1995-XX-XX	Y
1	*C227*	*PT-desc*	*com mais de dois mil anos*		
1	*C227*	*PT-narr*	*por mais de 2000 anos*		
1	C231	PT-desc	em Outubro de 1995	1995-10-XX	M
1	C231	PT-narr	em 1995	1995-XX-XX	Y
1	C236	PT-desc	em Novembro de 1995	1995-11-XX	M
1	C244	PT-title	1994	1994-XX-XX	Y
1	C244	PT-desc	em 1994	1994-XX-XX	Y
1	C244	PT-narr	Em 1994	1994-XX-XX	Y
1	C246	PT-desc	em 1995	1995-XX-XX	Y
1	C246	PT-narr	em 1995	1995-XX-XX	Y
2	*C266*	*PT-narr*	*nos dias de hoje*		
3	C326	PT-desc	de 1995	1995-XX-XX	Y
3	C327	PT-desc	em 1995	1995-XX-XX	Y
3	C327	PT-narr	em 1995	1995-XX-XX	Y
3	C332	PT-desc	em Novembro de 1994	1994-11-XX	M
3	C332	PT-narr	em Novembro de 1994	1994-11-XX	M
3	C334	PT-desc	em 1994	1994-XX-XX	Y
3	C334	PT-narr	em 1994	1994-XX-XX	Y
3	C335	PT-narr	em 1994	1994-XX-XX	Y
3	C336	PT-desc	durante 1994 e 1995	1994-XX-XX; 1995-XX-XX	Y
3	C337	PT-desc	de 1994	1994-XX-XX	Y
3	C339	PT-narr	em Dezembro de 1993	1993-12-XX	M
3	C340	PT-desc	em setembro de 1994	1994-09-XX	M
3	C340	PT-narr	em setembro de 1994	1994-09-XX	M
3	C341	PT-narr	em Fevereiro de 1994	1994-02-XX	M
3	C343	PT-desc	de 1994	1994-XX-XX	Y
3	C344	PT-desc	de 1994	1994-XX-XX	Y
3	C345	PT-desc	de 1994	1994-XX-XX	Y
3	C345	PT-narr	de 1994	1994-XX-XX	Y
3	C346	PT-desc	em 1995	1995-XX-XX	Y
3	C346	PT-narr	em 1995	1995-XX-XX	Y
3	C347	PT-title	1994	1994-XX-XX	Y
3	C347	PT-desc	em 1994	1994-XX-XX	Y
3	C348	PT-desc	em 1994	1994-XX-XX	Y
3	C349	PT-desc	em 1994	1994-XX-XX	Y
3	C350	PT-desc	em 1994	1994-XX-XX	Y

Table D.2: Temporal Information of the CHAVE Topics.