Disambiguating Implicit Temporal Queries for Temporal Information Retrieval Applications

Ricardo Nuno Taborda Campos
Doutoramento em Ciência de Computadores
Departamento de Ciência de Computadores
2013

Orientador
Gaël Dias, Professor Catedrático, Universidade de Caen Basse-Normandie

Coorientador
Alípio Mário Jorge, Professor Associado, Faculdade de Ciências da Universidade do Porto

Ricardo Nuno Taborda Campos

Disambiguating Implicit Temporal Queries for Temporal Information Retrieval Applications

Tese submetida à Faculdade de Ciências da Universidade do Porto
para obtenção do grau de Doutor em Ciência de Computadores

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
Março / 2013

To my grandparents Franklim and Berta who are always on my mind.

To my parents Arlindo and Suzel for their love, support, encouragement and teachings I was given.

To my sister Carla for believing in me ever since.

To my wife Célia for her fellowship and unconditional support. Thanks for being always with me.

To my little daughter Margarida for her love.

Acknowledgments

Like all good projects in our life, this PhD was a challenging pursuit that has taught me countless lessons. I will try not to forget some of them in the future:

Do daily backups of your information; Backups are useless when facing a tsunami. But don't panic. 90% of the tsunamis occurring in Hawaii are false alarms [reference unknown]; Criticisms improve our work; Having a work rejected is part of the learning process and the first step to have it accepted later; Doing research can be distressing as we are never satisfied with the end result (only when our papers get accepted); The PhD is a work in progress. Tomorrow is another day; Doing research is a unique opportunity to meet interesting and intelligent people from everywhere (and to make us believe that somehow we may also share that intelligence). Finally: we're nothing when we stand alone, which leads me to thank a number of people.

First and foremost, I would like to thank my supervisor Gaël Dias and my co-supervisor Alípio Jorge. I know Gaël since I was an under-graduate student. He has been a good friend and a reference with a strong impact on the way I teach and do research. I am deeply grateful for his lessons, ideas and for the discussions we had. I would also like to thank Alipio, without whom the writing of this thesis would have been far more difficult. His scientific rigor, his high level of demand and the motivation he places upon his work has made me a better researcher.

A special gratitude goes to Eugénio de Almeida, President of the Polytechnic Institute of Tomar, Maria da Conceição Fortunato, Director of the School of Management of Tomar and José Ribeiro Mendes, Director of the Information Communication and Technologies Department for giving me all the necessary conditions to pursue this Phd. I also have the fortune to work with Célio Marques and Vasco Silva who were always there whenever I needed. I am sincerely grateful to them.

I could not forget to thank Francisco Couto, Luis Rodeia, Maria João Simões, Pawan Lingras and Rajendra Akerkar for their support at a very initial stage.

Moreover, I must thank the Portuguese Government through FCT – The Foundation for Science and Technology (grant SFRH/BD/63646/2009) for the financial support without which this research could never have been completed, as well as the LIAAD - INESC TEC, the Center of Mathematics of University of Beira Interior, the Association for Computing Machinery and the European Summer School in Information Retrieval who supported my participation in several international conferences.

I would also like to acknowledge the Hultig (University of Beira Interior) staff, namely David Machado, Sebastião Pais and Isabel Marcelino for their support, and both Alexandra Ferreira from the Computer Science Department of FCUP, Paula Marques from the postgraduate office of FCUP and Pedro Almeida from LIAAD - INESC TEC for their kind assistance in administrative matters. A special thanks goes to José Moreno from the GREYC CNRS Laboratory (University of Caen) without whom I would be unable to install the web services.

I had the fortune to meet Fernando Diaz and Ian Ruthven, my two mentors of the Doctoral Consortium in SIGIR 2011. I would like to thank both for the useful discussions we had.

Many thanks to those anonymous conference reviewers whose very insightful comments have greatly improved my work. Regarding this, I could not forget Adam Jatowt who has offered very useful comments about related research studies.

I am especially grateful to some people for their efforts in evaluating the different stages of the system. In particular, I would like to thank Daniel Dias, Dário Ferreira, Miguel Baptista, Ricardo Bichinho, Rui Valdemar, Sandra Ferreira and Vasco Fernandes. I would also like to thank all my friends, especially Claudia Santos, Hugo Sainhas, Luis Nina, Ruben Pedro and Sandra Pinto for constantly encouraging me.

Finally, I would like to thank my family. Bento Nunes, Lucinda Nunes and Isabel Nunes for their kind support and understanding; Franklim Taborda and Berta Taborda, my grandparents who, wherever they are, will certainly be proud of my endeavors; Arlindo Campos and Suzel Campos, my parents, for the stability provided, for the values I was given, and for all the love and encouragement. Thanks for helping me make this dream come true; Carla Campos, my sister, for her support and motivated words during our long conversations. Her optimism undoubtedly contributed to achieve this goal. José Rodrigues for making me believe this was possible; Célia

Nunes, my wife, for her love and patience, for supporting my decision to pursue an academic career and for believing in me since the beginning. Our frequent and very productive discussions provided me with valuable insights about how to conduct this research. Many aspects of this research would not have been possible without her. I am forever grateful to her; Margarida Campos, my little daughter, who was always there for me with her big smile and without ever complaining for my long absences. Her caring, patience, and unconditional love have ensured the conclusion of this thesis.

Abstract

Time is an important dimension of the information retrieval area that can be very useful in helping to meet the users' information needs whenever they include temporal intents. However retrieving the information that meets the query demands is not an easy process. The ambiguity of the query is traditionally one of the causes impeding the retrieval of relevant information. This is particularly evident in the case of temporal queries where users tend to be subjective when expressing their intents (e.g., "*avatar movie*" instead of "*avatar movie 2009*"). Determining the possible times of the query is therefore of the utmost importance when attempting to achieve better disambiguated results and in order to enable new forms of exploring them.

In this thesis, we present our contributions to disambiguate implicit temporal queries in real-world environment, i.e. the Web. To understand better this type of queries, three directions may be followed: information extracted from (1) metadata, (2) query logs or (3) document contents. Within the context of this thesis, we will focus on the latter. However, unlike existing approaches we do not resort to a classification methodology. Instead, in our approach, we seek to detect relevant temporal expressions based on corpus statistics and a general similarity measure that makes use of co-occurrences of words and years extracted from the contents of the documents. Moreover, our methodology is language-independent as we do not use any linguistic-based techniques. Instead, we use a rule-based model solution supported by regular expressions.

Based on this, we start by performing a comprehensive study of the temporal value of web documents, particularly web snippets, showing that this type of collection is a valuable data source in the process of dating implicit temporal queries. We then develop two methods. A temporal similarity measure to evaluate the correlation between the query and the candidate dates identified, called Generic Temporal Evaluation (*GTE*) and a threshold-based classifier that selects

the most relevant dates while filtering out the non-relevant or incorrect ones, known as *GTE-Class*. Subsequently, we propose two different applications named *GTE-Cluster* and *GTE-Rank*. The first one, uses the determined time of the queries to improve search results exploration. For this purpose, we propose a flat temporal clustering model solution where documents are grouped at the year level. GTE-Rank, in turn, uses the same information to temporally re-rank the web search results. We employ a combination approach that considers words and temporal scores, where documents are ranked to reflect the relevance of the snippet for the query, both in the conceptual and in the temporal dimension.

Through extensive experimental evaluation, we mean to demonstrate that our models offer promising results in the field of Temporal Information Retrieval (*T-IR*), as demonstrated by the experiments conducted over web corpora. As an additional contribution to the research community, we publicly provide a number of web services so that each of the different approaches can be tested. Although the main motivation of our work is focused on queries with temporal nature, the implemented prototypes allow the execution of any query including non-temporal ones. Finally, for future research direction, we study the behavior of web snippets in the context of Future Information Retrieval (*F-IR*), a fairly recent topic which consists of extracting future temporal information in order to answer user queries with a future temporal nature.

Keywords

Temporal Information Retrieval, Temporal Query Understanding, Implicit Temporal Queries, Temporal Clustering, Future Information Retrieval, Temporal Web Mining.

Resumo

No contexto da pesquisa de informação, o tempo é uma dimensão que pode ser bastante útil para ajudar a satisfazer as necessidades de informação do utilizador com intenções temporais. No entanto, devolver a informação que o utilizador necessita não é um processo simples, sendo a ambiguidade da *query* uma das razões que tradicionalmente impede a obtenção de dados relevantes. Esta situação é particularmente evidente no caso de *queries* temporais onde os utilizadores tendem a ser subjetivos ao expressar as suas intenções (e.g. "*avatar movie*" em vez de "*avatar movie 2009*"). A determinação dos vários tempos associados a uma *query* assume assim grande importância na desambiguação dos resultados e na obtenção de novas formas de exploração dos mesmos.

Neste trabalho apresentamos uma proposta para desambiguar *queries* implicitamente temporais, em ambiente Web. Contrariamente às abordagens existentes, a nossa proposta não faz uso de metadados ou *query logs*, em vez disso, detetamos expressões temporais relevantes, com base nas estatísticas dos conteúdos dos documentos, e numa medida de similaridade que faz uso de coocorrências entre palavras e anos. A nossa abordagem é independente da língua dado que não é usada nenhuma técnica linguística, em vez disso, é utilizada uma solução baseada em regras (*rule-based*) assente na definição de expressões regulares.

Começamos por conduzir um estudo do valor temporal de documentos Web, nomeadamente, *web snippets*, mostrando que este tipo de coleções constitui uma valiosa fonte de informação no processo de datar *queries* implicitamente temporais. De seguida, desenvolvemos uma medida de similaridade temporal, denominada *Generic Temporal Evaluation* (*GTE*), para avaliar a correlação entre a *query* e o conjunto de datas candidatas identificadas; e um

classificador baseado em *threshold*, designado por *GTE-Class*, com a função de selecionar as datas mais relevantes e eliminar as datas não relevantes ou incorretas.

Posteriormente, propomos duas aplicações denominadas *GTE-Cluster* e *GTE-Rank*. A primeira, usa o(s) ano(s) da *query*, previamente determinado(s), para melhorar a exploração dos resultados. Para atingir este objetivo, propomos uma solução assente em *flat clusters* temporais onde os documentos são agrupados por ano. *GTE-Rank*, por seu lado, usa a mesma informação para reorganizar temporalmente os resultados da pesquisa. Desta forma, empregamos uma abordagem combinada onde os documentos são organizados de forma a refletir a relevância do *snippet* para com a *query*, tanto na dimensão concetual como na dimensão temporal.

Os resultados obtidos permitem concluir que os nossos métodos melhoram significativamente a performance das atuais abordagens. Para permitir que cada um dos diferentes algoritmos seja testado disponibilizamos um conjunto de *web services*. Embora a motivação principal do nosso trabalho esteja focada em *queries* de natureza temporal, os protótipos implementados permitem a execução de qualquer tipo de *query* incluindo *queries* não temporais.

Finalmente, como perspetiva de trabalho futuro, estudamos o comportamento dos *web snippets* no contexto da pesquisa de informação futura, um tópico relativamente recente que consiste na extração de informação temporal futura para permitir responder a *queries* desta natureza.

.

Palavras-Chave

Pesquisa de Informação Temporal, Entendimento Temporal da Pesquisa, Pesquisas Implicitamente Temporais, Agrupamento Temporal de Resultados, Pesquisa de Informação Futura, Mineração de Informação Temporal

Table of Contents

Acknowledgments **7**

Abstract **11**

Resumo **13**

Table of Contents **15**

List of Figures **19**

List of Tables **23**

Acronyms **27**

Notation **31**

1 Introduction **35**

1.1 Context 36

1.2 Problem Definition 38

1.2.1 Research Questions 39

1.2.2 Research Hypothesis 41

1.2.3 Research Objectives 41

1.3 Contributions 42

1.3.1 Scientific Contributions 42

1.3.2 Contributions to the Research Community 44

1.4 Evaluation 46

1.5 Thesis Structure 46

2 Temporal Information Extraction **49**

2.1 Models of Temporal Annotation of Documents 49

2.1.1 Definition of Time 49

2.1.2 Time and Timelines 50

2.1.3 Temporal Expressions 51

2.1.4 Temporal Information Extraction 52

2.2 Temporal Information in Web Resources 55

2.2.1 The Metadata-based Approach 55

2.2.2 The Content-based Approach 56

2.2.3 The Usage-based Approach 57

2.3 Extracting Temporal Information: Our Rule-based Model Solution 58

2.3.1 The Google Insights Dataset 58

2.3.2 The AOL Dataset 61

2.4 Summary 63

3 Dating Implicit Temporal Queries **65**

3.1 The Temporality of Web Snippets 66

3.2 Implicit Temporal Query Classification 70

3.3 Comparing the Temporal Value of Web Snippets with Web Query Logs 76

3.4 Summary 78

4 Temporal Disambiguation of Queries **79**

4.1 Related Research 81

4.2 Identifying Query Relevant Temporal Expressions 82

4.2.1 Web Search 83

4.2.2 Web Snippet Representation 84

4.2.3 GTE: Temporal Similarity Measure 85

4.2.4 GTE-Class: Date Filtering 92
4.3 Experimental Setup 94
4.3.1 Dataset Description 94
4.3.2 Baseline Measures 97
4.3.3 Evaluation Metrics 98
4.4 Results and Discussion 99
4.4.1 Experiment A 99
4.4.2 Experiment B 110
4.5 Summary 111
5 Temporal Clustering **113**
5.1 Related Research 114
5.2 GTE-Cluster 115
5.3 Results and Discussion 117
5.3.1 Experiment A 118
5.3.2 Experiment B 124
5.3.3 Experiment C 127
5.4 Summary 130
6 Temporal Re-Ranking of Web Search Results **131**
6.1 Related Research 133
6.2 GTE-Rank 134
6.3 Experimental Setup 136
6.3.1 Dataset Description 137
6.3.2 Baseline Methods 138
6.3.3 Evaluation Metrics 138
6.4 Results and Discussion 142
6.4.1 Experiment A 143

6.4.2 Experiment B 145
6.4.3 Experiment C 156
6.5 Summary 169
7 Future Information Retrieval **171**
7.1 Related Research 172
7.2 Results and Discussion 173
7.2.1 Experiment A 173
7.2.2 Experiment B 180
7.3 Summary 187
8 Conclusions and Future Research **189**
8.1 Future Research 190
References **193**

List of Figures

Figure 1.1: Result of Internet Archive for Yahoo! website. 36
Figure 1.2: Timeline of JFK assassination. 37
Figure 1.3: Predictions about climate change. 37
Figure 1.4: Google Book Ngram viewer for Albert Einstein and Sherlock Holmes phrases. 37
Figure 2.1: Timeline for the "*Haiti earthquake*" query. 50
Figure 2.2: Temporal document annotation model. 53
Figure 3.1: Distribution of dates from the Q450R100 dataset. 68
Figure 3.2: Distribution of dates per category from the Q450R100 dataset. 69
Figure 3.3: "*toyota recall*" query timeline for the "*1998-2011*" time span. Q450R100 dataset. 69
Figure 3.4: Snippet vs.. Title scatter plotter. 70
Figure 3.5: Fleiss Kappa values when varying θ for the $TQC(q)$ function 73
Figure 3.6: Yahoo! and Google query suggestion for the query "*bp oil spill*" 76
Figure 3.7: $Tsnippet(q)$ vs.. $TLogYahoo(q)$ scatter plot. 77
Figure 3.8: $Tsnippets(q)$ vs. $TLogGoogle(q)$ scatter plot. 77
Figure 4.1: GTE overall architecture. 83
Figure 4.2: Example of first-order and second-order similarity measures. 87
Figure 4.3: Context vector representations: (W;W); (D;D); (W;D); (D;W); (WD;WD). 88
Figure 4.4: (WD;WD) context vector representation for *Port-au-Prince* and *2010*. 90
Figure 4.5: Google suggestion for the query "*avatar movie*". 96
Figure 4.6: $IS(w_j,d_j)=0$. 101

Figure 4.7: Size and threshold effect. Median, Mean and Max/Min approach. Point biserial correlation values. 102
Figure 4.8: Recall, Precision and F1-M performance when varying λ for the BGTE. 105
Figure 4.9: ROC curve for the BGTE measure. 105
Figure 4.10: BGTE vs. Baselines. 107
Figure 5.1: Relevant GTE-Clusters retrieved for the query "*true grit*" 120
Figure 5.2: Non-relevant GTE-Clusters not retrieved for the query "*true grit*" 121
Figure 5.3: Survey results for the set of 42 queries. 128
Figure 5.4: GTE-Cluster interface for the query "*true grit*". Extracted from http://wia.info.unicaen.fr/GTEAspNetFlatTempCluster_Server 129
Figure 6.1: Top-10 results retrieved from Google for the query "*football world cup Germany*". 132
Figure 6.2: MAP. GRank1 vs. GRank2. $0.0 \leq \alpha \leq 1.0$. Top/Tail. WCRank_DS1 dataset. Solid markers indicates statistically significant improvement of the results of GRank1 over the GRank2 method using matched paired one-sided t-test with p-value < 0.05. 144
Figure 6.3: MRP. GRank1 vs. GRank2. $0.0 \leq \alpha \leq 1.0$. Top/Tail. WCRank_DS1 dataset. Solid markers indicates statistically significant improvement of the results of GRank1 over the GRank2 method using matched paired one-sided t-test with p-value < 0.05. 144
Figure 6.4: MAP. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Top. WCRank_DS1 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test 146
Figure 6.5: MRP. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Top. WCRank_DS1 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test 146
Figure 6.6: MRR. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Top. WCRank_DS1 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test 147

Figure 6.7: Average precision difference histogram for the 38 queries. GRank ($\alpha = 0.8$) vs. Baselines. Top. WCRank_DS1 dataset. 150
Figure 6.8: Average recall-precision for the 38 queries. GRank ($\alpha = 0.8$) vs. Baselines. Top. WCRank_DS1 dataset. 150
Figure 6.9: MAP. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Tail. WCRank_DS1 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test 151
Figure 6.10: MRP. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Tail. WCRank_DS1 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test 151
Figure 6.11: Average precision difference histogram for the 38 queries. GRank ($\alpha = 0.8$) vs. Baselines. Tail. WCRank_DS1 dataset. 153
Figure 6.12: Average recall-precision for the 38 queries. GRank ($\alpha = 0.8$) vs. Baselines. Tail. WCRank_DS1 dataset. 154
Figure 6.13: GTE-Rank interface for the query "*true grit*" over the WCRank_DS1. Extracted from http://wia.info.unicaen.fr/GTERankAspNet_Server 155
Figure 6.14: MAP. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Top. WCRank_DS2 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test 157
Figure 6.15: MRP. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Top. WCRank_DS2 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test 157
Figure 6.16: MRR. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Top. WCRank_DS2 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test 158

Figure 6.17: Average precision difference histogram for the 38 queries. GRank ($\alpha = 0.9$) vs. Baselines. Top. WCRank_DS2 dataset. 161
Figure 6.18: Average Recall-Precision for the 38 queries. GRank ($\alpha = 0.9$) vs. Baselines. Top. WCRank_DS2 dataset. 161
Figure 6.19: Interface of the GTE-Rank web service for the query "*true grit*" over the WCRank_DS2. Top 10 results. 162
Figure 6.20: MAP. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Tail. WCRank_DS2 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test 163
Figure 6.21: MRP. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Tail. WCRank_DS2 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test 163
Figure 6.22: Average precision difference histogram for the 38 queries. GRank ($\alpha = 0.8$) vs. Baselines. Tail. WCRank_DS2 dataset. 165
Figure 6.23: Average recall-precision for the 38 queries. GRank ($\alpha = 0.8$) vs. Baselines. Tail. WCRank_DS2 dataset. 167
Figure 6.24: GTE-Rank interface for the query "*true grit*" over the WCRank_DS2. Tail 10 results. Extracted from http://wia.info.unicaen.fr/GTERankAspNet_Server 168
Figure 7.1: Word cloud for near future dates. 179
Figure 7.2: Word cloud for distant future dates. 180
Figure 7.3: Overall analysis of global accuracy for snippet texts. 183
Figure 7.4: Text genre analysis for Naïve Bayes (D1,D2) and (D3,D4) comparison. 183
Figure 7.5: Overall analysis of global accuracy for title texts. 185
Figure 7.6: Text genre analysis for Multinomial Naïve Bayes (D1,D2) and (D3,D4) comparison. 185

List of Tables

Table 1.1: List of datasets and URLs [February 25th, 2013]. 44
Table 1.2: List of web services and URLs [February 25th, 2013]. 45
Table 1.3: List of user interfaces and URLs [February 25th, 2013]. 46
Table 2.1: List of query categories for the GISQC_DS dataset. 60
Table 2.2: Rule-based precision in the Q465R20, Q450R20 and Q450R100 datasets. 61
Table 2.3: List of query categories for the AOL_DS dataset. 62
Table 3.1: Average measure results in the Q465R20, Q450R20 and Q450R100 dataset. 67
Table 3.2: Pearson correlation between TTitle, TSnippet and TUrl. 70
Table 3.3: Query concept classification. Adapted from Song et al. [82]. 71
Table 3.4: Concept query classification of the Q450R100 dataset. 71
Table 3.5: Temporal ambiguity for the queries "*twilight eclipse*", "*toyota recall*", "*hdf netbanking*". 72
Table 3.6: Manual temporal query classification of the Q450R100 dataset. 73
Table 3.7: Confusion matrix representation. 74
Table 3.8: Stratified *5*-fold repeated random sub-sampling test dataset for the $TQC(q)$ function. 75
Table 3.9: Automatic temporal query classification of the Q450R100 dataset. 75
Table 3.10: Pearson correlation coefficient between TLogYahoo, TLogGoogle, TTitle, TSnippet, and TUrl. 77
Table 4.1: Running Example: Haiti earthquake. 85
Table 4.2: List of words that co-occur with *2010*. 86
Table 4.3: M_{ct} matrix for our running example. 90

Table 4.4: List of text queries. 94
Table 4.5: (q, d_j) classification for the query "*true grit*". 96
Table 4.6: Statistics of WC_DS and QLog_DS datasets. 96
Table 4.7: Best point biserial correlation coefficient for GTE. 101
Table 4.8: Point biserial correlation coefficient for GTE. $0 < T \leq 0.09$. N is fixed to $+\infty$. 101
Table 4.9: Point biserial correlation coefficient for GTE. $5 \leq N \leq +\infty$. T is fixed to *0.05*. 102
Table 4.10: Best point biserial correlation coefficient for the five context vectors. *T0.05*. 102
Table 4.11: List of classification (q, d_j) examples. BGTE vs. Baselines. 103
Table 4.12: Stratified *5*-fold repeated random sub-sampling test dataset. BGTE results. 104
Table 4.13: Comparative results for $sim(q, d_j)$. 106
Table 4.14: Comparative results for $F(sim(w_j, d_j))$, *F= Median*. 106
Table 4.15: Comparative results for $F(sim(w_j, d_j))$, *F= Arithmetic Median*. 108
Table 4.16: Comparative results for $F(sim(w_j, d_j))$, *F= Max/Min*. 108
Table 4.17: Best overall classification for each group of measures. 109
Table 4.18: BGTE vs. Baseline rule-based model. 110
Table 4.19: BGTE vs. Google_QLogs and Yahoo_QLogs. 111
Table 5.1: List of GTE-Clusters (left hand side) vs. non-GTE clusters (right hand side). 118
Table 5.2: (S_i, $d_{j,i}$) classification for the query "*true grit*". 122
Table 5.3: GTE-Cluster vs. non-GTE performance based on 656 distinct (S_i, $d_{j,i}$) pairs. Boldface indicates statistically significant improvement of the GTE-Cluster method compared with the non-GTE one using matched paired one-sided t-test with p-value < 0.05. 123
Table 5.4: Clustering evaluation of GTE-Cluster and Carrot over the WC_DS dataset. 125
Table 5.5: Snippet evaluation of GTE-Cluster and Carrot over the WC_DS dataset. 125
Table 5.6: Cluster list of the GTE-Cluster. 126
Table 5.7: Cluster list of Carrot search engine. 126
Table 5.8: User survey for the query "*true grit*". 127
Table 6.1: List of text queries. 136
Table 6.2: Relevance judgments for the WCRank_DS1 and WCRank_DS2 datasets. 137
Table 6.3: Recall and precision values for ranking from two queries. 140
Table 6.4: Average recall-precision at standard recall levels using interpolation. 141
Table 6.5: GTE-Rank experiments. 143

Table 6.6: MAP. GRank1 vs. GRank2. $0.0 \leq \alpha \leq 1.0$. Top/Tail. WCRank_DS1 dataset. Boldface indicates statistically significant improvement of the results of GRank1 over the GRank2 method using matched paired one-sided t-test with p-value < 0.05. 145
Table 6.7: MRP. GRank1 vs. GRank2. $0.0 \leq \alpha \leq 1.0$. Top/Tail. WCRank_DS1 dataset. Boldface indicates statistically significant improvement of the results of GRank1 over the GRank2 method using matched paired one-sided t-test with p-value < 0.05. 145
Table 6.8: MAP, MRP, MRR, P@1, P@3, P@5, NDCG@5, NDCG@10 and NDCG@20 results. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Top approach. WCRank_DS1 dataset. The absence of underline indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test. 147
Table 6.9: P@k, NDCG@k, MAP, MRP and MRR results. GRank vs. Baselines. Top approach. WCRank_DS1 dataset. The absence of underline indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test. 149
Table 6.10: Precision/Recall curve GRank ($\alpha = 0.8$) vs. Baselines. Top approach. WCRank_DS1 dataset. 150
Table 6.11: MAP, MRP, P@1, P@3 and P@5 results. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Tail approach. WCRank_DS1 dataset. All the comparisons are statistically significant with p-value < 0.05 using the matched paired one-sided t-test. 152
Table 6.12: P@k, MAP and MRP results. GRank vs. Baselines. Tail approach. WCRank_DS1 dataset. All the comparisons are statistically significant with p-value < 0.05 using the matched paired one-sided t-test. 153
Table 6.13: Precision/Recall. GRank ($\alpha = 0.8$) vs. Baselines. Tail. WCRank_DS1 dataset.. 154
Table 6.14: MAP, MRP, MRR, P@1, P@3, P@5, NDCG@5, NDCG@10 and NDCG@20 results. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Top approach. WCRank_DS2 dataset. The absence of underline indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test. 159

Table 6.15: P@k, NDCG@k, MAP, MRP and MRR results. GRank vs. Baselines. Top approach. WCRank_DS2 dataset. The absence of underline indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test. 160
Table 6.16: Precision/Recall curve GRank ($\alpha = 0.9$) vs. Baselines. Top approach. WCRank_DS2 dataset. 161
Table 6.17: MAP, MRP, P@5, P@10 and P@20 results. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Tail approach. WCRank_DS2 dataset. All the comparisons are statistically significant with p-value < 0.05 using the matched paired one-sided t-test. 164
Table 6.18: P@k, MAP and MRP results. GRank vs. Baselines. Top approach. WCRank_DS2 dataset. All the comparisons are statistically significant with p-value < 0.05 using the matched paired one-sided t-test. 165
Table 6.19: Precision/Recall curve GRank ($\alpha = 0.8$) vs. Baselines. Tail approach. WCRank_DS2 dataset. 167
Table 7.1: Web snippets future temporal value. 176
Table 7.2: Number of queries resulting in the retrieval of web snippets with future dates. 176
Table 7.3: Classification of texts according to genre. 179
Table 7.4: Classification of texts according to genre for near and distant future dates. 179
Table 7.5: Datasets structure. 181
Table 7.6: Snippet classification results for the boolean and tf-idf cases. 182
Table 7.7: Title classification results for the boolean and tf-idf cases. 184
Table 7.8: Snippet clustering results for the K-means in the boolean and tf-idf cases. 186
Table 7.9: Title clustering results for the K-means in the boolean and tf-idf cases. 187

Acronyms

ACE	:	Automated Content Extraction
AOL_DS	:	AOL Log Dataset
AP	:	Average Precision
AUC	:	Area Under Curve
BA	:	Balanced Accuracy or Efficiency
BGTE	:	Best Generic Temporal Evaluation
BRank	:	Bing Rank
CBC	:	Clustering by Committee
DCG	:	Discounted Cumulative Gain
F1-M	:	F1-Measure
F-IR	:	Future Information Retrieval
FN	:	False Negative
FP	:	False Positive
GISQC_DS	:	Google Insights for Search Query Classification Dataset
GTE	:	Generic Temporal Evaluation
GTE-Class	:	Generic Temporal Evaluation Classification model
GTE-Cluster	:	Generic Temporal Evaluation Clustering model
GTE-Rank	:	Generic Temporal Evaluation Ranking model
GRank	:	GTE Rank
HAC	:	Hierarchical Agglomerative Clustering
IR	:	Information Retrieval
IS	:	InfoSimba

LSA	:	Latent Semantic Analysis
NER	:	Named-entity Recognition
NDCG	:	Normalized Discounted Cumulative Gain
NGD	:	Normalized Google Distance
NIST	:	National Institute of Standards and Technology
NYT	:	New York Times
MAP	:	Mean Average Precision
MIT	:	Massachusetts Institute of Technology
MRP	:	Mean R-Precision
MRR	:	Mean Reciprocal Rank
MUC	:	Message Understanding Conference
ORank	:	Ordered Rank
P	:	Precision
P@k	:	Precision at k documents
PMI	:	Pointwise Mutual Information
QLog_DS	:	Query Log Dataset
R	:	Recall or Sensitivity
R@k	:	Recall at k documents
ROC Curve	:	Receiver Operating Characteristic Curve
RP	:	R-Precision
RR	:	Reciprocal Rank
RRank	:	Random Rank
Rule-based model	:	Temporal information tagger supported on regular expressions
SCP	:	Symmetric Conditional Probability
Spec	:	Specificity
SVM	:	Support Vector Machine
T-IE	:	Temporal Information Extraction
T-IR	:	Temporal Information Retrieval
TAC	:	Text Analysis Conference
TAIA	:	Time-Aware Information Access Workshop
TERN	:	Time Expression Recognition and Normalization
TN	:	True Negative

TP	:	True Positive
TWAW	:	Temporal Web Analytics workshop
WC_DS	:	Web Content Dataset
WCRank_DS	:	Web Content Rank Dataset

Notation

$\boldsymbol{q}$	:	A query
$\boldsymbol{TLogYahoo(q)}$	:	The temporal value of the Yahoo! auto-completion engine
$\boldsymbol{TLogGoogle(q)}$	:	The temporal value of the Google auto-completion engine
$\boldsymbol{TTitle(q)}$	:	The title temporal value of the set of web snippets retrieved for the query q
$\boldsymbol{TTitle(.)}$	:	The average of $TTitle$ for all the queries
$\boldsymbol{TSnippet(q)}$	:	The descriptive text temporal value of the set of web snippets retrieved for the query q
$\boldsymbol{TSnippet(.)}$	:	The average of $TSnippet$ for all the queries
$\boldsymbol{TUrl(q)}$	:	The link temporal value of the set of web snippets retrieved for the query q
$\boldsymbol{TUrl(.)}$	:	The average of $TUrl$ for all the queries
$\boldsymbol{TA(q)}$	:	Temporal ambiguity query function model, which aims to determine the temporal aggregated value of $TTitle(q)$, $TSnippet(q)$ and $TUrl(q)$
$\boldsymbol{TQC(q)}$	:	Temporal query classification model, which aims to determine whether a query q is or not temporal
$\boldsymbol{FutureDates(q)}$	:	The future temporal value of the texts retrieved (titles, snippets or URLs) for the query q
$\boldsymbol{FutureDates(.)}$	:	The average of $FutureDates$ for all the queries
$\boldsymbol{NearFuture(q)}$	:	The near or distant future temporal value of the texts retrieved (titles, snippets or URLs) for the query q

$\boldsymbol{NearFuture}(.)$	:	The average of $NearFuture$ for all the queries
$\boldsymbol{d_j}$	:	A temporal pattern, i.e. a candidate year. May or may not be a date
$\boldsymbol{S}$	:	The set of n web snippets retrieved in response to the query q
$\boldsymbol{S_i}$	:	A single snippet
$\boldsymbol{W_S}$	:	The set of distinct relevant words/multiwords, i.e. the relevant vocabulary, extracted for a query q, within the set of web snippets S
$\boldsymbol{W_{S_i}}$	:	The set of the k most relevant words/multiwords associated with a web snippet S_i
$\boldsymbol{w_{h,i}}$	:	$w_{h,i} \in W_{S_i}, h = 1,..,k$ is one of the k most relevant words/multiwords of the snippet S_i.
$\mathbf{W_{d_j}}$	:	The set of words that appear together with the candidate date d_j, in any web snippet S_i from S
$\boldsymbol{W^*}$	:	The set of distinct words that results from the intersection between the set of words W_S and the set of words W_{d_j}
w_j	:	A word/multiword of the set W^*
$\boldsymbol{D_S}$	:	The set of distinct candidate years extracted for a query q within the set of web snippets S
$\boldsymbol{D_S^{Rel}}$	:	The set of relevant years extracted for a query q within the set of web snippets S
$\boldsymbol{D_{S_i}}$	:	The set of t candidate years associated with a web snippet S_i
$\boldsymbol{d_{j,i}}$	:	$d_{j,i} \in D_{S_i}$, $j = 1,..,t$, is one of the t candidate dates of the snippet S_i
$\#\boldsymbol{Rel}$	:	Number of $d_{j,i}$ whose relevance judgments equals to 1
$\#\overline{\boldsymbol{Rel}}$	:	Number of $d_{j,i}$ whose relevance judgments equals to 0
$\boldsymbol{D_{S_i}^{Rel}}$	:	The set of u relevant years associated with a web snippet S_i
$\boldsymbol{d_{j,i}^{Rel}}$	:	$d_{j,i}^{Rel} \in D_{S_i}^{Rel}, j = 1,..,u$ is one of the u relevant dates of the snippet S_i
$\boldsymbol{GTE(q, d_j)}$	:	The temporal similarity between a query q and a candidate year d_j

$\boldsymbol{V_{GTE_{D_s}}}$: The vector that stores the temporal similarity between the candidate date d_j and the query q for the t distinct candidate dates

$\boldsymbol{V_{GTE_{D_s}^{Rel}}}$: The matrix that stores the temporal similarity between relevant dates d_j and the query q for the m distinct relevant dates

$\boldsymbol{M_{ct}}$: Conceptual temporal correlation matrix that stores the DICE, PMI or SCP similarities between "*word*"-"*word*", "*candidate date*"-"*candidate date*" and "*word*"-"*candidate date*"

$\boldsymbol{M_{CT}^{Rel}}$: Conceptual temporal correlation matrix that stores the DICE, PMI or SCP similarities between "*word*"-"*word*", "*date*"-"*date*" and "*word*"-"*date*"

$\boldsymbol{IS_(X;Y)_S_F}$: Notation for the different versions of the GTE. *IS* means Infosimba, *(X;Y)* means the representation type of the context vectors, *S* the similarity measure used in IS (PMI, SCP and DICE), whose values are registered in M_{ct} and *F* is the aggregation function that combines the different similarity values between $w_j \in W^*$ and d_j

$\boldsymbol{Sim(S_i, S_j)}$: Determine the similarity between the snippet S_i and the snippet S_j

$\boldsymbol{Cluster\text{-}Rank(C_j, S_i)}$: Determine the ranking position of the snippet S_i within the cluster C_j

$\boldsymbol{GTE\text{-}Rank(q, S_i)}$: Determine the ranking position of the snippet S_i with regard to the query q

Chapter 1

Introduction

The World Wide Web (*WWW*) is currently a large information network, where the retrieval and the organization of results with relevant and quality content remains an open question for mostly all ambiguous queries. In this context, the inclusion of a temporal dimension can play an important role in increasing the quality of the retrieved results. With this in mind, traditional commercial search engines try to provide means to perform web search based on time. Notwithstanding, little effort has been done to incorporate temporal features in their architectures. Indeed, in most cases, systems are limited to simply asking the user to explicitly specify a time span. Thus, even for retrieval systems that work quite well, the quality of the results of some queries is poor. For example, when querying "*Iraq war*", most search engines will mainly retrieve results from the last Iraq war, when the user may be interested not only in "*2003*" but also in "*1991*". One reason for this lies in the difficulties that exist in relating the temporal information found in the documents with the implicit intentions of the user's query. Example 1.1 shows an example for the query "*world cup*".

> Miss Universe was held this year in Bahamas. *2008* was an incredible year, but everybody is waiting for the FIFA South Africa Football **World Cup**.

Example 1.1: Associated years for the query "*world cup*".

It is evident from this text that "*2008*" is not related with the query. Actually, the FIFA South Africa Football World Cup was only held in 2010. Such limitations lead to loss of precision and less relevant retrieval information, making it difficult to have a time perspective associated with temporal queries. Understanding the timeline of the documents and the query is

therefore of the utmost importance and potentially useful for several tasks. In what follows, we describe some of the most important applications developed so far.

1.1 Context

Time is an inherent construct to human life as our thinking is often defined in the form of chronologically arranged events stretching from past, to present and future. Many information needs have underlying temporal intent(s). For example, users may require documents describing the past (e.g. queries about biographies of historical persons), documents containing the most recent, up-to-date information (e.g. queries about weather or currency rate) or even future-related information (e.g. queries about planned events in a certain geographical area). Temporal Information Retrieval (*T-IR*) is an emerging area of research that takes into account the temporal dimension in the retrieval of information needs. In general, T-IR aims to satisfy these temporal needs and combine traditional notions of *document relevance* with the so-called *temporal relevance*. This would enable the retrieval of temporally-relevant documents and a temporal overview of search results in the form of timelines or similar visualization structures.

Some efforts have been made in the past few years and a number of temporal applications have been developed. One of the first initiatives is the Internet Archive project [52] that aims to build a digital library of websites. The objective is to store different versions of websites based on their timely updates. Figure 1.1 shows an example for the URL www.yahoo.com.

Figure 1.1: Result of Internet Archive for Yahoo! website.

There is also much research on using temporal information for exploration and search purposes. For instance, the Massachusetts Institute of Technology (*MIT*) has developed SIMILE Timeline Visualization[1] project, a web widget prototype for visualizing temporal data as shown in Figure 1.2 about the event on the assassination of John Fitzgerald Kennedy.

[1] http://www.simile-widgets.org/timeline [February 25th, 2013]

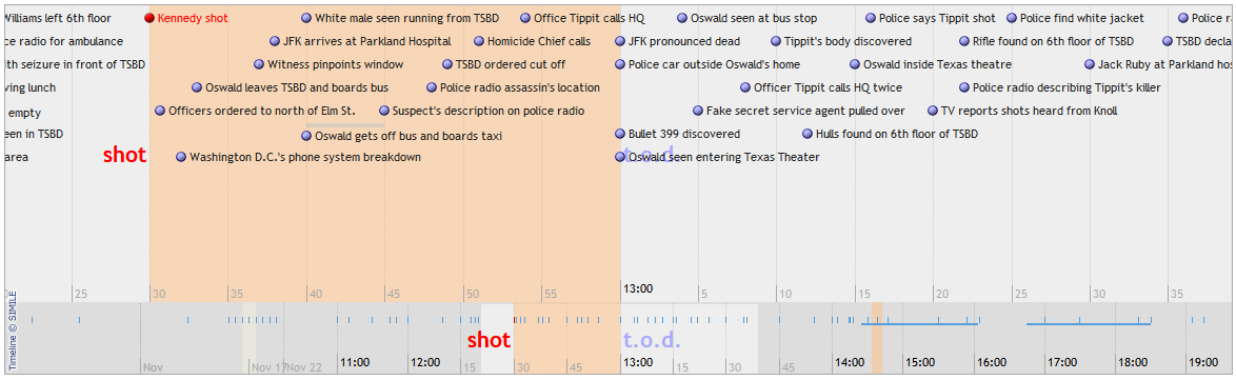


Figure 1.2: Timeline of JFK assassination.

Recorded Future[2] and Yahoo!'s via its Time Explorer [63] application (see Figure 1.3) have also been working on some specific analysis tools concerning the retrieval of future-related information.

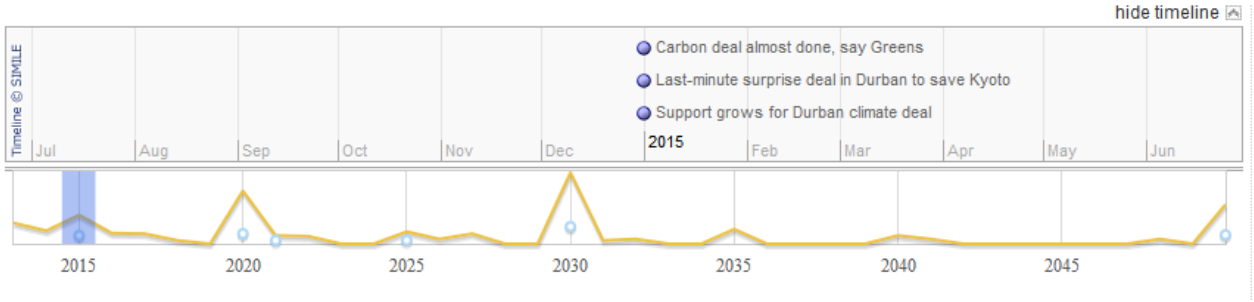


Figure 1.3: Predictions about climate change.

Google has also recently introduced the Google NGram Viewer[3] (see Figure 1.4) a visualization tool that shows the rises and falls of particular keywords across 5 million books over selected years. All these huge projects clearly evidence the importance of T-IR as a new promising research area.

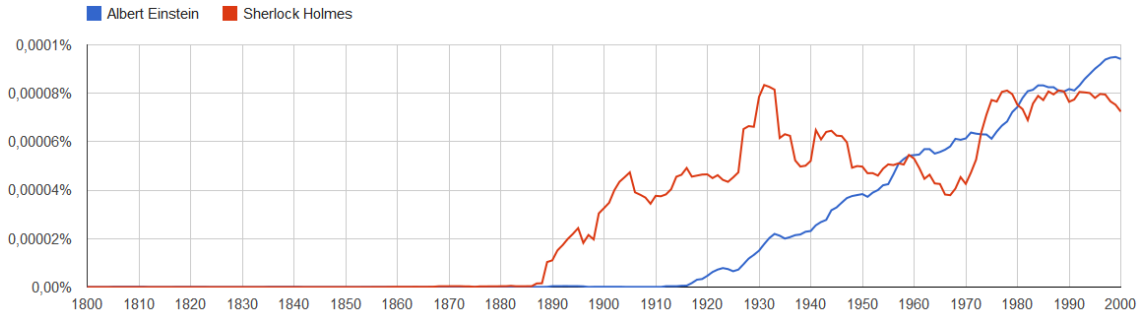


Figure 1.4: Google Book Ngram viewer for Albert Einstein and Sherlock Holmes phrases.

[2] http://www.recordedfuture.com [February 25th, 2013]
[3] http://books.google.com/ngrams [February 25th, 2013]

Another evidence of the importance of T-IR is the organization of contests and workshops focusing on temporality. For the former, different competitions have been proposed, such as the Message Understanding Conference (*MUC*) with specific tracks on the identification of temporal expressions (MUC6 and MUC7), the Automated Content Extraction (*ACE*) evaluation program, organized by National Institute of Standards and Technology (*NIST*) and recently attached to the Text Analysis Conferences (*TAC*), the Time Expression Recognition and Normalization (*TERN*) and the TempEval within the SemEval competition. As an example of the latter we may take the WWW Temporal Web Analytics workshop (*TWAW* 2011, 2012 and 2013) or the SIGIR Time-Aware Information Access workshop (*TAIA* 2012). This has lead to the creation of annotation standard corpora like the TimeBank [75], annotation schemas such as TimeML[4] [74] and the development of temporal taggers, later discussed on Section 2.1.4.

Based on all these factors, an upsurge of applications is expected in the near future, mostly concerning temporal information exploration, new forms of search results exploration, but also applications concerning micro-collections (e.g. blogs, twitter posts). In particular various research studies have already been proposed in different sub-areas of T-IR. The work of Ricardo Baeza-Yates in 2005 [8] defines the foundations of T-IR. Then different works have been tackled in several topics, such as user query understanding [10, 31, 51, 57, 65, 85], temporal web snippets generation [3, 6], temporal ranking of documents [11, 30, 38, 40, 53, 88] temporal clustering [2, 5], future retrieval [8, 48, 49] or temporal web image retrieval [36]. A more detailed categorization of the relevant research carried out in this research area, can be found in a Wikipedia webpage created for this purpose, named *Temporal Information Retrieval*[5]. In the following section we describe the main objectives of our work.

1.2 Problem Definition

The aim of this thesis is to explore the temporal dimension so to enhance the results as well as the presentation of information retrieval operations[6]. In what follows we lay down our research questions, research hypothesis and objectives.

[4] http://www.timeml.org/site/index.html [February 25th, 2013]
[5] http://en.wikipedia.org/wiki/Temporal_information_retrieval [February 25th, 2013]
[6] This chapter is partially based on the work published at the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval – SIGIR 2011 (Campos 2011).

1.2.1 Research Questions

The main research question of this thesis is how to date implicit temporal queries in a way that allows us to reach the temporal disambiguation of the query "on-the-fly". We are particularly interested in understanding the temporal nature of any given implicit temporal query - i.e. queries with an inherent temporal intent not explicitly defined by the user - so as to improve temporal ranking and temporal clustering of results as well as future information retrieval. To approach this problem in a more adequate manner we provide a more detailed account of the problem by setting up a few more questions below. We divide them into questions regarding content and query analysis, query temporal disambiguation and temporal information retrieval models.

Content Analysis

The extraction of temporal information plays an important role in the process of dating implicit temporal queries. However, finding reliable information is not an easy process. First, we should guarantee that the information extracted is trustworthy and query-related. Then, we must ensure that it is up-to-date and available for extraction. After confirming these assumptions, we propose to extract temporal information from the contents of the web documents. This is in contrast with the extraction of temporal information within the timestamp of the document or the query log, which may not be able to offer either trustworthy or available information. Having this defined, we formulate our first research question:

Q1. Do web sources have enough temporal value to date implicit temporal queries?

Query Analysis

Temporal queries can be divided into explicit and implicit. Explicit temporal queries are those tagged with an explicit timestamp (e.g. "*football world cup 2010*"), whereas implicit temporal ones are those for which no time has been explicitly assigned and yet have a temporal nature (e.g. "*football world cup*"). In this thesis, we are particularly interested in dealing with the latter. Subsequently, we wish to estimate how frequent are queries with an implicit temporal nature. While, Metzler et al. [65] have already estimated this value based on information extracted from web query logs, no one, to the best of our knowledge, has performed a similar study based on information extracted from web documents. Thus, the second research question we address is:

Q2. How many queries have an implicit temporal nature?

Query Temporal Disambiguation

In this thesis we are particularly interested in dating implicit temporal queries. Instead of using the document timestamp or web query logs, we want to use temporal information extracted from the documents contents. However, given that a web document can have countless temporal references, we need to judge which ones are query relevant and which ones are not. Based on this, we formulate two further research questions:

Q3. How to model the relations between the query and the different times found within the web documents?

Q4. How to identify the most relevant dates and subsequently remove the non-relevant ones?

Temporal Information Retrieval Models

In general, temporal information is provided by means of timelines. An alternative to this commonly used interface is to present results based on temporal clusters. Thus the fifth research question we address is:

Q5: How to use temporal clusters to temporally disambiguate the most relevant time periods of the query?

Another possibility yet consists of re-ranking web documents according to the user's query temporal intent. This leads us to the sixth question:

Q6: How to combine conceptual and temporal relevance in re-ranking models when no temporal criterion is provided in the query?

Finally, we want to study future temporal references extracted from web documents so as to realize whether we can help identifying and understanding the future temporal nature of an implicit temporal query or not. Specifically, we set up a classification and a clustering task, which is aimed at identifying the nature of future-related texts, i.e., informative, scheduled or rumor, based on data features extracted from web documents. Thus, the two last research questions addressed in this thesis are:

Q7. How does future-related information in web documents impact the text classification of future-related texts?

Q8. How does future-related information in web documents impact the clustering of future-related texts?

1.2.2 Research Hypothesis

Bearing in mind the questions posed, we will now define the research hypothesis:

H1. Web documents incorporate a high level of temporal information compared with available web query logs;

H2. There is a significant difference between temporally classifying a query based on information extracted from the contents of the web documents or from web query logs;

H3. Our temporal similarity measure to evaluate the degree of relation between a query and a candidate date, enables us to better identify the most relevant dates related to the query;

H4. The introduction of a classification model that is able to identify top relevant dates for any given implicit query while filtering out non-relevant ones, improves the correct classification of a query and a candidate date pair when compared to the baseline approach, which considers all the candidate dates as relevant for the query;

H5. The combination of our classification model with a clustering methodology, allows for a better identification of the most relevant time periods of the query;

H6. A linear combination of the conceptual relevance with the determined time(s) of the query enhances the temporal nature of the web search results;

H7. Temporal features detected in web documents improve the predictive ability of correctly classifying future-related texts into one of the three following categories: informative, scheduled or rumor;

H8. Temporal features improve the clustering precision of texts containing references to future events;

In the following section we present the research objectives in more detail.

1.2.3 Research Objectives

We start by studying the temporal characteristics of web documents and compare them to web query logs to make sure that web documents are reliable when dating queries with an inherent temporal nature. Then, we investigate the number of queries that have an inherent implicit temporal intent, so as to estimate the focus target of this thesis. Next, we show how to determine the correct time intents of implicit temporal queries and its effect in the retrieval effectiveness.

This step is of the utmost importance as it gathers information that is used as input during the subsequent processes, namely the clustering and the ranking of search results. Following this, we assess whether clustering and re-ranking results get effectively improved with the introduction of the determined relevant time of the query. The following objective is to assess whether web documents can be used for future analysis. Finally, we identify the nature of future texts to understand how temporal features may impact the classification and clustering of its different types, i.e. informative, scheduled and rumor.

1.3 Contributions

Our research produced some scientific contributions as well as datasets and web services for the research community. In this section we present the main ones.

1.3.1 Scientific Contributions

Our research extracts temporal information from web snippets and investigates how this information can be used to improve query understanding and search results exploration. In what follows, we present a summary of our contributions. We make reference to the corresponding contribution question and indicate the chapters where further details can be found.

C1 We provide new measures to understand and compare the temporal value of web snippets and web query logs. In addition, we determine which of the two data sources retrieves a wider range of different dates.
[*Related to Q1, which will be further discussed in Chapter 3*]

C2 We perform the first study to determine the number of queries having an implicit temporal nature upon information extracted from web snippets. In particular, we define a temporal ambiguity function and a query classification model to help determining whether a query is or not temporal.
[*Related to Q2, which will be further discussed in Chapter 3*]

C3 We elaborate a temporal similarity measure called *GTE* which evaluates the degree of relation between candidate dates and a given query based on a second-order attributional similarity metric. We compare the results of GTE with first order similarity measures and with the baseline rule-based model (current standard in most of the T-IR tasks), which

selects all of the temporal patterns found as correct dates. To accomplish this, we resort to a statistical measure that particularly suits this task.
[*Related to Q3, which will be further discussed in Chapter 4*]

C4 We propose the employment of the *GTE-Class*: a classification model that is able to identify top relevant dates for any given implicit query and to filter out non-relevant ones. In this regard, we propose two different methods, one based on a threshold classification and a further one based on the application of a machine learning algorithm. To conduct both experiments we rely on classical IR metrics.
[*Related to Q4, which will be further discussed in Chapter 4*]

C5 Similarly to the work of Alonso et al. [5] we design a flat temporal clustering solution, called *GTE-Cluster*, to group search results by time based on web snippets sharing the same year. We propose to integrate our temporal classification model in order to correctly identify relevant temporal clusters and snippet members for the query. Finally, we compare our clustering proposal with current web snippet clustering engines and conduct a user study to test the performance of our approach on a real web user environment. For both experiments we rely on classical IR metrics.
[*Related to Q5, which will be further discussed in Chapter 5*]

C6 In line with what has been proposed by Kanhabua et al. [53] this study defines a novel temporal ranking model, called *GTE-Rank*, that takes into account both content importance and temporal distance to re-rank web snippets. In particular, we study the impact of the incorporation of our temporal classification model into the retrieval effectiveness and propose a set of measures that will enable us to test not only how GTE-Rank performs when pulling relevant documents to the top, but also when pushing down non-relevant ones.
[*Related to Q6, which will be further discussed in Chapter 6*]

C7 Finally, this research measures the future temporal nature of web documents and assesses the impact of using temporal features in the classification and clustering of the different types of future-related texts. We propose two measures for the first step and apply traditional algorithms for the classification and clustering steps.
[*Related to Q7* and *Q8, which will be further discussed in Chapter 7*]

1.3.2 Contributions to the Research Community

We publicly provide a set of queries and ground-truth results to the research community. Hence, our evaluation results can be compared to future approaches. The first dataset is called *GISQC_DS* and was constructed with a twofold purpose: (1) to enable to study the temporal value of web snippets and (2) to enable an account of the percentage of queries having a temporal nature. The second dataset is called *WC_DS* and was designed to evaluate the relation between (*query*, *candidate date*) pairs and (*web snippets*, *candidate date*) pairs. The third dataset, *GISFD_DS*, was developed to supply a set of (*web snippets*, *future candidate dates*) pairs. The fourth and fifth datasets, named *QLog_DS* and *AOL_DS*, were meant to provide query-log resources. While the first one is based on Google and Yahoo! auto-completion search engines, the second one is a sample of a previously available release of AOL search engine[7]. Finally the sixth dataset called *WCRank_DS* was developed to provide a graded relevance between (*query*, *web snippets*) pairs. A list of all datasets is provided in Table 1.1.

Table 1.1: List of datasets and URLs [February 25th, 2013].

Name	Description	URL
GISQC_DS	Google Insights for Search Query Classification Dataset	http://www.ccc.ipt.pt/~ricardo/datasets/GISQC_DS.html
WC_DS	Web Content Dataset	http://www.ccc.ipt.pt/~ricardo/datasets/WC_DS.html
GISFD_DS	Google Insights for Search Future Dates Dataset	http://www.ccc.ipt.pt/~ricardo/datasets/GISFD_DS.html
QLog_DS	Query Log Dataset	http://www.ccc.ipt.pt/~ricardo/datasets/QLog_DS.html
AOL_DS	AOL Log Dataset	http://www.ccc.ipt.pt/~ricardo/datasets/AOL_DS.html
WCRank_DS	Web Content Rank Dataset	http://www.ccc.ipt.pt/~ricardo/datasets/WCRank_DS.html

In addition, we make available a number of web services, so that each of the proposals can be tested by the research community. In order to retrieve the query results, we rely on the recently launched Bing Search API[8] parameterized with the *en-US* market language parameter to retrieve 50 results per query. The proposed solutions are computationally efficient and can easily be tested online. While the main motivation of our work is temporal queries, we show that our methods are robust enough to handle atemporal ones as well. Below is a detailed description of each web service.

[7] http://www.aol.com/ [February 25th, 2013]
[8] https://datamarket.azure.com/dataset/5BA839F1-12CE-4CCE-BF57-A49D98D29A44 [February 25th, 2013]

- **GTE_1** returns, in XML format, the GTE similarity value calculated between the query and all the candidate dates together with the corresponding contents - i.e. title, snippet and url - where the set of candidate dates appear. It can also be understood as the **GTE_1-Cluster** web service, where the similarity value corresponds to the similarity between the query and the respective temporal cluster (given by the candidate date).
- **GTE_2** returns, in XML format, the GTE similarity value calculated between the query and all the dates classified by the GTE-Class as relevant. In addition, it returns the set of contents - i.e. title, snippet and url - where the set of relevant dates appear. It can also be understood as the **GTE_2-Cluster** web service, where the similarity value corresponds to the similarity between the query and the respective temporal cluster (given by the relevant date).
- **GTE-Class** returns, in XML format, those dates classified by the GTE-Class as relevant for the query.
- **GTE-$Rank_1$** returns, in XML format, the set of fifty re-ranked web snippets.
- **GTE-$Rank_2$** returns, in XML format, a filter of the re-ranked web snippets containing only relevant dates.

A complete list of all web services is given in Table 1.2. Note that, in order to work, each web service should be added a query at the end of the URL.

Table 1.2: List of web services and URLs [February 25th, 2013].

Name	URL
GTE_1 GTE_1-Cluster	http://wia.info.unicaen.fr/GTEAspNetFlatTempCluster_Server/api/GTE?FilterDates=false&query=
GTE_2 GTE_2-Cluster	http://wia.info.unicaen.fr/GTEAspNetFlatTempCluster_Server/api/GTE?FilterDates=true&query=
GTE-Class	http://wia.info.unicaen.fr/GTEAspNetFlatTempCluster_Server/api/GTE?query=
GTE-$Rank_1$	http://wia.info.unicaen.fr/GTERankAspNet_Server/api/GTERank?AllSnippets=false&query=
GTE-$Rank_2$	http://wia.info.unicaen.fr/GTERankAspNet_Server/api/GTERank?AllSnippets=false&query=

Furthermore, we provide two user interfaces so that the research community can test the GTE-Cluster and the GTE-Rank applications. Below is a description of both:

- **GTE-Cluster** user interface, which offers the user two options: to return all the clusters (including the non-relevant ones) or to return only the relevant ones;
- **GTE-Rank** user interface, which offers the user two options: to return all the web snippets (including those not having dates) or to return only the web snippets with relevant dates.

The two URLs are given in Table 1.3.

Table 1.3: List of user interfaces and URLs [February 25th, 2013].

Name	URL
GTE-Cluster	http://wia.info.unicaen.fr/GTEAspNetFlatTempCluster_Server
GTE-Rank	http://wia.info.unicaen.fr/GTERankAspNet_Server

Finally, this research also concurs to the current knowledge on this subject with the creation of a Wikipedia webpage named *Temporal Information Retrieval* which categorizes relevant research carried out in the context of T-IR. The url for this web page has already been provided in Section 1.1.

1.4 Evaluation

We conduct a variety of experiments to empirically evaluate the effectiveness of our approaches. For this purpose, we have used appropriate statistical tests to assess the validity of the proposed solutions. Tests were complemented with the application of traditional Information Retrieval (*IR*) metrics or the definition of new ones, when appropriate.

1.5 Thesis Structure

In this thesis, each objective laid out above is addressed by defining different temporal models. For each one, we present the challenges posed and describe the set of experiments undertaken. It is important to note that we introduce related research in each chapter instead of presenting it in a classical dedicated chapter. The remainder of this thesis is structured as follows.

Chapter 2: Temporal Information Extraction establishes the fundamental notions of extracting temporal information from text documents and presents them in accordance with our rule-based model solution.

Chapter 3: Dating Implicit Temporal Queries investigates the temporal value of web snippets and query logs in order to assess if they can be used to date queries with an inherent temporal nature. We define a set of basic metrics to represent the temporal value of each of the two collections and a temporal ambiguity measure, complemented with a temporal query classification model that enables us to automatically classify a query with regard to its temporal value (*Temporal*, *ATemporal*).

Chapter 4: Temporal Disambiguation of Queries establishes the foundations of our approach, which will serve as the basis for the rest of this thesis. In this context, we introduce the overall theoretical framework known as *GTE* and present our approach named *GTE-Class* to identify relevant dates to text queries.

Chapter 5: Temporal Clustering details our flat temporal clustering algorithm, called *GTE-Cluster,* which was tested under real web user environment.

Chapter 6: Temporal Re-Ranking of Web Search Results presents a new re-ranking algorithm called *GTE-Rank* showing the effectiveness of our approach under the variation of different parameters.

Chapter 7: Future Information Retrieval discusses whether web snippets can be used to understand the future temporal nature of text queries and describes the results of applying classification and clustering algorithms to group informative, schedule and rumor texts. The techniques discussed shed light on how temporal features impact upon the classification and clustering of future-related web documents.

Chapter 8: Conclusions and Future Research presents a general overview of the improvements achieved by this thesis and suggests future research trajectories.

Chapter 2

Temporal Information Extraction

Before retrieving temporally relevant documents for a given query, we must identify and normalize temporal expressions found in documents. In this chapter we describe the foundations of Temporal Information Extraction (*T-IE*) and present our rule-based model solution. More specifically, Section 2.1 gives an overview of document temporal annotation models. Section 2.2 describes the different number of approaches that can be used to extract time features within web collections. Section 2.3 describes and evaluates the rule-based model solution used in this thesis to extract temporal features. Finally, Section 2.4 summarizes this chapter.

2.1 Models of Temporal Annotation of Documents

We shall now introduce the main concepts and definitions of T-IE and highlight some of the main problems underlying the methodologies we employ. Section 2.1.1 provides an operational definition of Time. Section 2.1.2 shows the underlying relation between time and timelines. Section 2.1.3 underlines the different types of temporal expressions. Finally, Section 2.1.4 outlines the process of extracting temporal information from texts.

2.1.1 Definition of Time

In a simpler way, Time can be defined as an ongoing sequence of events. Each instance of time is a point-in-time value, commonly referred to as *chronon* [4], an indivisible unit that cannot be further divided into new temporal points. Commonly, a chronon can assume eight different instances, from the coarsest to the finest significant granularity: century (*c*), decade (*de*), year (*Y*),

quarter (*q*), semester (*s*), month (*M*), week (*w*) and day (*D*). Note that a date can also include any other time points, such as hours, minutes, seconds, a fraction of a second and so forth.

Time values can be physically represented in a calendar, a timekeeping system by which time is organized into several different granularities. Following the ISO-8601:2004[9] standard, a date in the Gregorian calendar is usually represented in the form of *YYYY-MM-DD*, where [*YYYY*] indicates a four-digit year, [*MM*] indicates a two-digit month, [*DD*] indicates a two-digit day of that month. Although not very common, a date representation can also include the week number. In this case, the month is replaced by the corresponding week, which results in the format *YYYY-Www-DD*, where *ww* means the week number, from *W01* to *W52*. Moreover, the Gregorian calendar can have a number of different specialized calendars, such as fiscal, sports, business or academic ones.

When addressing the time issue within the scope of database applications, two types of time are considered: *focus time* and *transaction time*. While focus time is related to the period of time in which events have occurred in real life, i.e. the time of the fact itself, the transaction time is that specific time when the fact is stored in a database. In the web context, the focus time would be the time mentioned in the content of web pages, while transaction time would be its timestamp, i.e. the point in time when the document was created, modified or published. A more in-depth discussion on this topic is given in Section 2.2. In the next section we show how time can be represented in a timeline.

2.1.2 Time and Timelines

The sequence of events is usually represented in a timeline. A timeline, also known as chronology, is a graphic representation listing important events of a query within a particular time span. An example of a timeline is what a user would construct to represent the history of the Haiti earthquake query as shown in Figure 2.1.

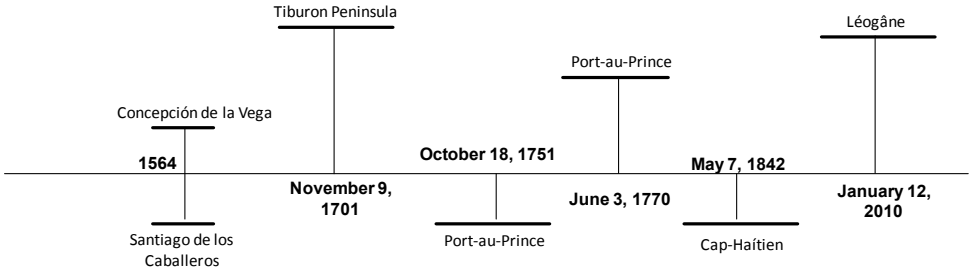


Figure 2.1: Timeline for the "*Haiti earthquake*" query.

[9] http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=40874 [February 25th, 2013]

Depending on their purpose, timelines of different granularities can be constructed, either more fined-grained (e.g. t_q for quarters, t_s for semesters, t_M for months, t_w for weeks and t_D for days) or more coarse-grained (e.g. t_c for centuries, t_{de} for decades and t_Y for centuries). In this thesis we define T_E^g of a query q as a timeline, where E is the set of events of the query and g the granularity. In the case of our example, q = {*Haiti earthquake*}, E = {*Concepción de la Vega*,.., *Leógâne*} and g = {t_Y, t_M, t_D}. In what follows we describe the different types of temporal expressions.

2.1.3 Temporal Expressions

Temporal expressions are a very rich form of natural language that can be defined as a sequence of tokens with temporal meaning. This includes dates (e.g. *2013-12-25*), but also other types of temporal references such as time adverbs (e.g. "*yesterday*"), propositional phrases (e.g. "*on Monday*"), verbs (e.g. "*opened five years ago*") or nouns (e.g. "*January*", "*summer"*). According to the formal specification language for time data TimeML [74], temporal expressions can be classified into three categories. Depending on the type of anchoring process they operate, they can be organized as according to:

- The Duration;
- The Set;
- The Time/Date.

To be more precise, the *Duration*, provides information about the length of an interval (e.g. "*he has been playing for* <TIMEX>***5 years***</TIMEX>"). The *Set* provides data about the periodicity or frequency of the temporal instance (e.g. "*he plays* <TIMEX>***twice a week***</TIMEX>"). Finally, the *Time/Date* refers to a specific chronon, a unique point-in-time in the timeline (e.g. "*the game will take place at* <TIMEX>***4pm***</TIMEX> *on* <TIMEX>***25 of December 2012***</TIMEX>").

The greatest difficulty in developing an automatic system for detecting temporal expressions is the infinite diversity of ways in which time can be expressed. As such, temporal expressions can be further structured into three other types according to their temporal reference. Following the work of Alonso et al. [4] we distinguish between:

- Explicit Temporal Expressions;
- Implicit Temporal Expressions;
- Relative Temporal Expressions.

Explicit temporal expressions were first referenced [79] in 1995 during MUC-5 [1]. They denote a precise moment in the timeline and can be determined without further knowledge. Based on the granularity level, we may have for example "*2009*" for the year granularity, "*December 2009*" for the month and "*2012.12.25*" for the day.

Implicit expressions are often associated with events carrying an implicit temporal nature. They are very difficult to position in time mostly due to the inexistence of a clear temporal purpose or a clear unambiguous associated time point. For example, expressions such as "*Christmas day*", embody a temporal nature not explicitly specified. Therefore, as pointed out by Alonso et al. [3] they require that at least a year chronon appears close to the event in order to relate them to their correct temporal value. For example, "*miss universe*" could be normalized to "*2012.12.19*", if we refer to the contest of Miss Universe which took place on September 2012.

Relative temporal expressions were referenced for the first time [79] in 1998 during MUC-7 [24]. They depend on the document publication context. For instance, the expressions "*today*", "*last Thursday*" or "*45 minutes after*" are all relative to the document timestamps or to the nearby absolute dates. As such, finding the document timestamp is of the utmost importance so that the expression may be mapped directly on the timeline as an explicit expression. An example of this would be the normalization process of the expression "*today*" into the document creation time "*2012.12.19*". While this kind of information is usually available in news documents, it is particularly difficult to find it within web documents, as we will discuss in Section 2.2.1. Besides, having access to the document timestamp, however important, might not be enough in the case of more complex phrases. An example of this is the expression "*on Thursday*", which, as observed by Alonso et al. [6], can either refer to the previous, or to the next Thursday. In the following section we describe the process of extracting temporal information from documents.

2.1.4 Temporal Information Extraction

The identification of temporal information is a non-trivial task that requires a common pre-processing stage of the document usually involving four steps. The first step is the *Tokenization* which divides the text into words or phrases. The second step is *Sentence Extraction* which identifies the most relevant sentences in texts. The third step is *Part-of-Speech Tagging*, where tokens are assigned to morpho-syntactic information. Finally, the fourth step is Named-entity Recognition (*NER*), which involves the identification of proper names in the document, such as persons, locations and organizations.

Interestingly, temporal expressions have also been part of the NER process. However, since 2004, with the introduction of the TERN task as part of the ACE program, Temporal Information Extraction has become a separate independent task. As such, once the text processing is under way, the T-IE process can start.

More to the point, it consists of three main tasks. The first task is the *Recognition* of the temporal expressions. The second task is the *Normalisation* with the purpose of unifying the different ways in which temporal expressions can be expressed. Finally, the last task called *Temporal Annotation* aims to express temporal expressions in a standard format. The result is a set of texts annotated with temporal expressions. Figure 2.2 shows the whole process. It is important to notice that not all the pre-processing steps are necessary to perform temporal information extraction.

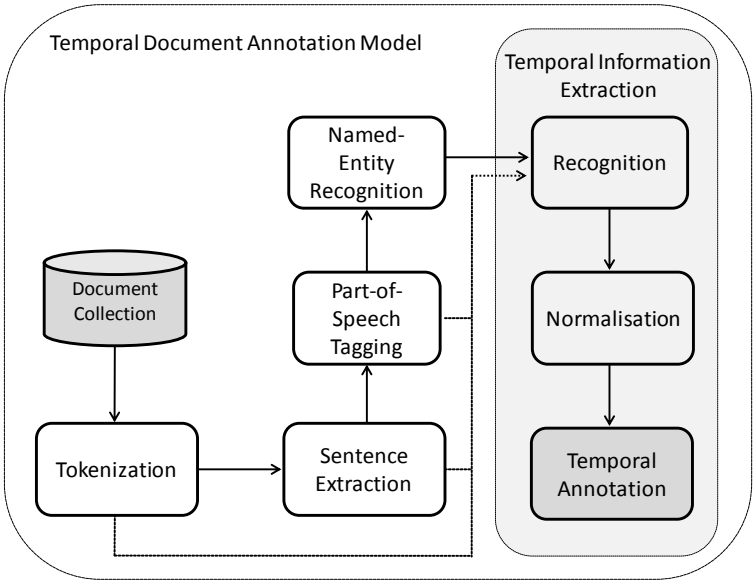


Figure 2.2: Temporal document annotation model.

The overall process of T-IE is usually conducted by temporal taggers, which follow *rule-based* approaches. These are based on regular expressions or local grammar-based techniques, usually involving hard work by experienced experts.

In the last few years, temporal taggers have become an important research area with several proposals. However, most of available temporal taggers are useful for only one language (typically English) and one domain (usually the news domain). In the following part, we offer a detailed account of four of the most known temporal taggers: TempEx [62], GUTime[10], Annie[11]

[10] http://www.timeml.org/site/tarsqi/modules/gutime/download.html [February 25th, 2013]
[11] http://www.aktors.org/technologies/annie/ [February 25th, 2013]

and HeidelTime[12] [84]. A more detailed description of existing approaches can be found in Strötgen & Gertz [84].

TempEx [62] was the first temporal expression tagger to be developed. It is a rule-based model that extracts temporal information, particularly explicit (e.g. "*December 24, 2009*") and relative temporal expressions (e.g. "*Monday*"), marked with TIMEX2 tags. First, the document is tokenized into words and sentences, and part-of-speech is used. Each sentence is then passed on to a module that identifies time expressions. The entire document is then passed on to a discourse processing module which resolves context-dependent time expressions such as indexicals. Tests were performed using news articles collections, from the New York Times, and Voice of America, ABC and CNN broadcasts, reporting 82.7% of recall, 83.7% of precision and 83.2% of F-measure.

GUTime was developed in 2002 under the supervision of the Georgetown University. It extends the capabilities of TempEx by adding TIMEX3 tags. For example, "*last week*" as referred by the authors could be represented not only by the time value but also by the week preceding the week of the document date. It was evaluated on the TERN 2004 training corpus achieving an F-Measure score of 85% and 82% for temporal expressions recognition.

Annie was also developed in 2002 as part of the GATE[13] distribution [29]. Dates are recognized by a NER system which consists of pattern-action rules. ANNIE has been adapted to Bulgarian, Romanian, Bengali, Greek, Spanish, Swedish, German, Italian, and French language.

More recently, *HeidelTime* [84] was developed as a multi-lingual temporal tagger (English, German and Dutch) adapted not only to the news domain but also to narrative documents. HeidelTime is developed as a rule-based system using the TimeML annotation standard to tag temporal expressions. Tested in the TempEval-2 challenge [83] it has achieved the best performance system with an F-Score of 86% for the extraction and an accuracy of 85% for the normalization.

Although there has been significant advances in temporal tagging, applying existing time-taggers to web collections, may still result in incorrect time classification, causing a negative impact on systems effectiveness. Example 2.1 shows the resulting document produced by the HeidelTime temporal tagger on a short text and its limitations as well. It is clear that most of the

[12] http://dbs.ifi.uni-heidelberg.de/index.php?id=form-downloads [February 25th, 2013]
[13] http://gate.ac.uk/download/index.html [February 25th, 2012]

errors are the result of an incorrect annotation that tends to misclassify, some of the four consecutive numbers detected.

> *PoemsOnly.com archive of 2011 of romantic love*
> In this Valentine's Day we offer you some special presents. LCD Monitor with *2010 x* 768px screen resolution by 1316USD (or *1000€*). Call (*1619*) *1819*-5407 for reservation or send us a mail to *1122* NW David, *2012*-006 Portland.
> PoemsOnly.com @*2010*/06/32

Example 2.1: HeidelTime temporal tagger example.

In the following section, we describe the different approaches that may be used to extract time features with regard to web document collections.

2.2 Temporal Information in Web Resources

The extraction of time features within web documents can be done following one of three approaches:

- Metadata-based;
- Content-based;
- Usage-based.

Each of these three methodologies is usually related to the type of collection used i.e. web posts (e.g. news articles, blog posts, tweets, wikis) for metadata-based approaches, web documents (e.g. web pages, web snippets) for content-based solutions and web query logs for usage-based methodologies. Each one of these is described in the following sub-sections.

2.2.1 The Metadata-based Approach

The *Metadata-based* approach extracts temporal information from the metadata of a document. This includes the document creation time, the document publication time and the last-modified date. But it may also embody the extraction of additional temporal information from the document structure, in particular, information extracted from the URL of the document or from the anchor text itself[14].

[14] Note that Metadata simply refers to structured information embedded in the web source excluding any reference to the content of the document. This is the typical definition used in the field of T-IR and should not be compared to the terminology used in digital libraries (e.g., Dublin Core).

This information may be extremely useful to solve relative temporal expressions found in the content of a document (e.g. "*today*") and normalize them with a concrete date (e.g. "*2012/12/31*"). However, it may be inadequate in many cases due to the fact that the time of a document (creation, modification or publication time) may differ significantly from its actual content, i.e. focus time. A simple example of this would be a document published in "*2009*" but whose content concerns the year "*2011*".

While this information can be easily extracted from web news articles, it is particularly difficult to achieve successful results in the case of less structured collections such as web pages. This is because web servers typically do not provide more temporal information than the crawling date as referred by Nunes et al. [68]. An alternative solution is to extract this information from the document content, for instance, any temporal expressions preceded by the phrase "*last-modified*". Knowing this procedure to be quite an easy one, nonetheless demands a rule definition for each different language, which is quite unfeasible for real world applications. Moreover, it is unclear whether this information is reliable, as valid last-modified values are estimated to range from 40% to 80% [68].

2.2.2 The Content-based Approach

The *Content-based* approach focus on the analysis and extraction of temporal features within the Web contents, i.e. the focus-time. This includes looking for information within web pages, web micro collections or web archives.

Unlike metadata-based approaches, this process implies an increased level of difficulty as it usually involves linguistic analysis of texts, as previously discussed in Section 2.1.4. However, as the web is heterogeneous, multi-lingual, multi-cultural and highly multi-domain, ambiguity is common. An illustrative example is the expression "*New year*" which refers to a different point in time in the USA or in China. The same expression can even be expressed in a number of different languages (e.g. "*New year*" in English and "*除夕*" in traditional Chinese). Other problems relate to multi-lingual time formats (e.g. "*December 31, 2012*" would be translated to "*31 de Dezembro de 2012*" in Portuguese). In this case, one should build a time-tagger for each language. Moreover, similarly to the application of part-of-speech taggers, one may face some problems when applying temporal taggers to micro collections, such as web snippets or tweets. Indeed, their application may eventually result in poor outcomes, mostly due to a lack of

background, which is inherent to the small number of characters allowed for this type of sources (e.g. 140 in tweet posts) and the specific language used to write these texts (e.g. "*tomorrow*" may be transcribed by "*tomoz*"[15]).

2.2.3 The Usage-based Approach

Finally, the *usage-based* approach considers the extraction of temporal information mainly within web query logs, which consist of flat sets of files that record server temporal activity in a twofold perspective:

- Query timestamp;
- Query content time.

Query timestamp is the timestamp of the query, i.e. the date when the query was issued. It is mostly used to understand changes in query popularity and changing intent. The second type, *Query content time* relates to the content time of the query, i.e. the time which the user's query refers to. This can be explicitly provided by the user in the query (e.g. "*football world cup 2010*"), or implicitly defined (e.g. "*football world cup*"). While in the case of explicit temporal queries, the temporal nature is defined at the outset, in the case of implicit temporal ones, that information is not available. One possible solution to retrieve the explicit temporal value, is to look for related information within query logs. However, query logs are difficult to be accessed outside big industrial labs due to privacy issues [14]. One example of this is the AOL collection consisting of 21,011,240 queries, which is officially not available anymore because of the Thelma Arnoid case pointed out by the journalist of the New York Times[16]. Moreover, queries are highly dependent on users' own intents. Indeed, the simple fact that a query is year-qualified does not necessarily mean that it has a temporal intent (e.g. "*microsoft office 2007*", "*HP 1430*") or that the associated year is correlated to the query (e.g. "*football world cup 2012*" – there was no world cup in 2012). Furthermore, while web content or metadata approaches simply requires the set of web search results, the query log-based solution is query-dependent as it implies that some versions of the query have already been issued. This problem becomes even worse when only a small fraction of queries have explicit temporal patterns, as will be demonstrated in Section 2.3.1. In the upcoming part, we describe the rule-based model solution used in this thesis to extract temporal features.

[15] http://en.wikipedia.org/wiki/SMS_language [February 25th, 2013]
[16] http://en.wikipedia.org/wiki/AOL_search_data_leak [February 25th, 2013]

2.3 Extracting Temporal Information: Our Rule-based Model Solution

The identification of dates, either within queries or web documents, is probably the most recognized area with TempEx, GUTime, ANNIE and HeidelTime temporal taggers taking the lead. In this research, we propose to use a simple rule-based model supported on regular expressions that extract explicit temporal dates following these simple patterns: *YYYY*, *YYYY-YYYY*, *YYYY/YYYY*, *MM/dd/YYYY*, *dd/MM/YYYY*, *MM.dd.YYYY* and *dd.MM/YYYY*.

In this thesis, we are particularly interested in working at the year granularity level in order to keep language-independence and allow longer timelines for visualization. As such, although it is possible to extract temporal expressions with finer granularities, such as months and days, we end up normalizing each temporal expression to the year granularity level. So, for each discovered pattern, the temporal expression is normalized to *YYYY*.

Note, however, that a document can also contain other types of temporal expressions, other than explicit ones. This includes implicit and relative temporal expressions. Nevertheless, these ones will not be studied in this thesis as they require linguistic pre-processing steps that lie outside the scope of this thesis.

In the following part, we test the precision (see Equation 2.1) of our rule-based model in a collection of web documents and queries, with True Positives (*TP*) being the number of years correctly identified and False Positives (*FP*) being the number of years wrongly identified. More details on the evaluation metrics will be given later in Section 3.2.

$$Precision\ (P) = \frac{TP}{TP+FP}. \tag{2.1}$$

To conduct our experiments we consider two different datasets which are publicly available: the GISQC_DS and the AOL_DS. Each one will be described below.

2.3.1 The Google Insights Dataset

The Google Insights for Search Query Classification dataset (GISQC_DS) consists of 540 queries extracted from Google Insights for Search, which registered the hottest queries performed worldwide[17]. The queries selected belong to the period between January 2010 and October 2010 and result of a manual selection of 20 queries per each of the 27 pre-defined available categories. After removing duplicates, we end up with a set of 465 queries, including 15 explicit temporal

[17] The Google Insights for Search closed on September 27, 2012.

ones. Most of the queries belong to the categories of Internet (12.69%), Computer & Electronics (9.89%) and Entertainment (7.96%). A list of all the categories with a detailed description is given in Table 2.1.

On December 2010 each of the 465 queries was then issued in Bing[18] and Yahoo![19] search engines, with the parameter "*Number of Results to Return*" set to 20 and 100, so as to observe any variations that may exist due to the retrieval of a different number of results. Then, duplicated search results were removed.

Next, we removed the set of 15 explicit temporal queries that were part of the initial set, thus forming a new set of 450 queries. This will enable us to study the temporal value of web snippets (later on Section 3.1) and the future temporal value of web snippets (later on Section 7.2.1) in response to the simple execution of implicit temporal queries. As a consequence, the final sets consist of three collections denoted *Q465R20*, *Q450R20* and *Q450R100*, where *Q* means the number of queries issued and *R* the number of results retrieved for each query.

The results of each query were then assessed with regard to the correctness of each temporal pattern found. With this goal in mind, we manually went through each of the web snippets of the three collections *Q465R20*, *Q450R20* and *Q450R100* and checked whether the identified years were correct dates or not[20]. It is important to note that this evaluation has been carried out by only one human judge as it is a simple non-ambiguous task. As such, annotation inter-agreement does not apply.

[18] http://www.bing.com [February 25th, 2013]
[19] http://www.yahoo.com [February 25th, 2013]
[20] Note that for a web snippet we mean its title, snippet (descriptive text of the web snippet) and its url.

Table 2.1: List of query categories for the GISQC_DS dataset.

Query Category	Description	Example	%
Internet	Downloads, Chats, Facebooks, Google	Chrome download	12.69%
Computer & Electronics	Software, Hardware, Technology	Windows 7	9.89%
Entertainment	TV, Radio, Movies, Series, Music, Journals	Lady gaga	7.96%
Business & Economics	Prices, Sale of Products, Enterprises	Jobs	7.74%
Other	Other things	Plan b	6.24%
Games & Toys	Games, Lottery, Online Games	Mario bros	6.02%
Sports	Football, Race Horses	Marathon	5.59%
Literature	Books, Culture, Translators	Urban dictionary	4.73%
Travel, Maps & Weather	Travel, maps, forecast	Google maps	4.30%
Real Estate & Classified	Real Estate, Agents	Rent	4.09%
Finance & Insurance	Banks, Money, Currencies, Forms, Taxes	Bank of america	3.66%
Automotive	Cars, Caravans, Bikes, Boats, Motorcycle	Dacia duster	3.44%
Education & Science	Schools, Research	Big bang theory	3.44%
Beauty & Personal Care	Hairstyles, Tattoos, SPAs	Tattoo	3.44%
Food & Drink	Recipes, Food, Restaurants	Pizza	3.01%
Home & Garden	Furniture, Utilities	Furniture	2.58%
Health	Diseases	Diabetes	2.15%
URL	Links	facebook.fr	1.29%
Society	Horoscopo, Babys, Names, Weddings	Names	1.29%
Photo & Video	Photos, Videos	Photography	1.29%
Dates	Queries explicitly related with dates	Calendar	1.29%
Animals & Nature	Animals, Nature	Paul octopus	1.08%
News & Events	News, Events	News	0.86%
Country & Places	Countries and Places	Las Vegas	0.86%
Politics	Issues related to Politics	Presidential elections	0.43%
Military & Security	Military, Security	Security	0.43%
Porn	Movies, SexShops, Utilities	Sex	0.21%

The primary conclusion of our study is that our rule-based model solution is capable of achieving on average for the three collections, 96.4% within titles, 94.4% of precision in detecting years within snippets correctly, but significantly less in the case of URLs (82.5%). Overall, false dates tend to occur in the response of queries belonging to the categories of Internet (e.g. "*1600 YouTube Videos*"), Computer & Electronics (e.g. "*1024 x 768*"), Games & Toys (e.g. "*1000 games*") and Food & Drink (e.g. "*1001 recipes*"). It is worth noting that these results come

from the simple detection of dates, so it is expectable that a large number of errors occur when taking date relevance into account. A summary of the results is given in Table 2.2 for the three different collections.

Table 2.2: Rule-based precision in the Q465R20, Q450R20 and Q450R100 datasets.

Collections	# of Web Snippets Retrieved	Rule-based model Precision	
Q465R20	16,648	Title	97.9%
		Snippet	95.8%
		URL	85.1%
Q450R20	16,129	Title	95.8%
		Snippet	94.3%
		URL	75.0%
Q450R100	62,842	Title	95.3%
		Snippet	93.1%
		URL	87.4%

2.3.2 The AOL Dataset

The AOL Log dataset (AOL_DS) consists of 21,011,240 queries extracted from a previous release of the AOL search engine. From this collection, we applied our rule-based model solution and automatically selected those queries marked with explicit temporal references (e.g. "*football world cup 2006*" or "*dacia 1465*"). We ended up with a set of 143,590 possible temporal explicit queries, which represent 1.41% of the entire collection in line with the 1.5% claimed by Nunes et al. [69]. Our next step is to estimate the effective number of explicit temporal patterns as some of the detected years may be misleading (e.g. "*dacia 1465*") and to categorize each of the queries as in the previous dataset, i.e., according to the categories listed in Table 2.1.

In order to make these tasks feasible, we selected a representative statistical sample of 601 queries, denoted *Q601*. To reach this number of queries, we relied on the work of Barbetta et al. [9] and defined a maximum tolerated average sampling error E, of 4%, for a confidence interval of 95% following Equation 2.2:

$$n = \frac{(z_{0.975})^2}{4E^2}, \tag{2.2}$$

where z_p, which in this case is equal to 1.96, is the p-th quantile of the normal distribution and n is the determined number of queries.

Each of the 601 queries was then manually classified into the set of 27 categories as in the GISQC_DS dataset. A large majority of the queries belong to the categories of Automotive (21.96%), Entertainment (9.48%) and Sports (8.15%). A list of all the categories with a detailed description is given in Table 2.3.

Table 2.3: List of query categories for the AOL_DS dataset.

Query Category	**Description**	**Example**	**%**
Automotive	Cars, Caravans, Bikes, Boats, Motorcycle	1500cc dune buggy	21.96%
Entertainment	TV, Radio, Movies, Series, Music, Journals	1080 fm radio	9.48%
Sports	Football, Race Horses	ncaa baksetball 2006	8.15%
Society	Horoscopo, Babys, Names, Weddings	1930 census holly	6.84%
Other	Other things	kqfa1170	6.49%
Business & Economics	Prices, Sale of Products, Enterprises	1829 german coins	5.99%
News & Events	News, Events	conference may 2006	5.16%
Computer & Electronics	Software, Hardware, Technology	HP 1430	4.49%
Military & Security	Military, Security	1970-1971 attacks	3.49%
Education & Science	Schools, Research	Atlas project 2010	3.16%
Dates	Queries explicitly related with dates	May 2006 calendar	3.00%
URL	Links	www.1800lastbid.com	3.00%
Politics	Issues related to Politics	election 2004	2.50%
Finance & Insurance	Banks, Money, Currencies, Forms, Taxes	2006taxlaws	2.33%
Games & Toys	Games, Lottery, Online Games	Trivia questions 1960	1.83%
Photo & Video	Photos, Videos	gray 1792 picture	1.83%
Animals & Nature	Animals, Nature	Hurricanes in 2004	1.16%
Beauty & Personal Care	Hairstyles, Tattoos, SPAs	1970's outfits	1.16%
Home & Garden	Furniture, Utilities	lane chests 1930s	1.16%
Literature	Books, Culture, Translators	top books for 2005	1.16%
Travel, Maps & Weather	Travel, maps, forecast	travel ireland 2006	1.16%
Country & Places	Countries and Places	American flag in 1943	1.00%
Food & Drink	Recipes, Food, Restaurants	food eaten in 1850's	0.83%
Internet	Downloads, Chats, Facebooks, Google	free windows 2000	0.83%
Health	Diseases	medicinal of 1600's	0.67%
Porn	Movies, SexShops, Utilities	1500 naked picture	0.67%
Real Estate & Classified	Real Estate, Agents	1031 properties	0.50%

Finally, we classified each query as to whether or not the temporal pattern found is a real date. For example, the query "*1500 naked pictures*" would be labeled as a false positive, whereas the query "*1829 german coins*" would be classified as a true positive occurrence. The obtained results, show that our rule-based model is capable of achieving a precision of 86% in correctly identifying real temporal patterns from queries. This means that 14% of the queries, mostly belonging to the category of Computer & Electronics (e.g. "*hp 1430*"), still contain incorrect temporal patterns. If we generalize these results to the overall collection, we can conclude that, unlike the 1.41% previously indicated, an even small fraction of 1.21% of the queries are combined with dates. This support the claims presented in Section 2.2.3 and show how difficult it is to adopt a usage-based approach.

2.4 Summary

In this chapter, we presented the fundamental definitions of the Temporal Information Retrieval research area, which will serve as a contextualization basis for the rest of the thesis. In particular, we formalized the definition of time and timelines, and we introduced the notion of temporal expressions. We also presented the temporal information extraction process and defined the different methodologies used to extract temporal information from the web. Finally, we introduced our rule-based model solution and evaluated its precision in detecting explicit temporal patterns correctly in a collection of web documents and queries. It is worth noting that most of the incorrect temporal patterns belong to the category of Computer & Electronics (e.g. "*nikon d3000*"). This may cause possible biased results in case of considering this information as a valid temporal feature. One way to overcome this is to model, with some degree of confidence, the relationship existing between the query topics and the temporal patterns found, in such a way as to identify the top relevant dates. This will deserve further discussion in Chapter 4.

In the next chapter we ask whether the temporal information found within a collection of web documents and query logs can be used to date implicit temporal queries.

Chapter 3

Dating Implicit Temporal Queries

Understanding the temporal nature of a query is one of the most interesting challenges in Temporal Information Retrieval as referred by Berberich et al. [10]. However, few studies have attempted to answer the question as to "*How many queries have a temporal intent?*" or more specifically, the question as to "*How many of them have an explicit/implicit temporal nature?*". If we are able to answer these questions, we may estimate how many queries are affected by a temporal approach. However, inferring this information is a hard challenge. Firstly, different semantic concepts or facets can be related to a query. Secondly, it is difficult to define the boundaries between what is temporal and what is not. Thirdly, even if temporal intents can be identified by human annotators, the question remains as how we can transpose this into an automatic process. One possible solution is to seek related temporal references over web examples. Hence, in this chapter[21], we study the temporal value of two web data sources. On the one hand, we enquire into web snippets as a collection of web search results for any given query. On the other hand, we explore Google and Yahoo! completion engines, which provide indirect query-log access in order to understand the users' temporal intents. Our goal is to investigate the usefulness of each of these sources in order to date implicit text queries. As a result of our investigation, we propose different measures to understand the temporal value of each of the two data sources and define a temporal ambiguity function and a query classification model to help determining whether a query is or is not temporal.

[21] This chapter is partially based on the work published at the 1st International Temporal Web Analytics Workshop associated with WWW2011 (Campos et al. 2011b) and the Query Representation and Understanding Workshop associated with SIGIR2011 (Campos et al. 2011c).

The chapter is structured as follows. Section 3.1 studies the temporal value of web snippets. Section 3.2 assesses the percentage of queries having a temporal nature. Section 3.3 compares the temporal value of web snippets with web query logs. Finally, Section 3.4 summarizes the results of our study.

3.1 The Temporality of Web Snippets

In this section, we are particularly interested in studying the existence of temporal information within web snippets. To the best of our knowledge, this is the first work towards a comprehensive data analysis having web snippets as a data source. For the first experiment, we considered three web collections, from the GISQC_DS dataset, named Q465R20 (20 results per query), Q450R20 and Q450R100 (100 results per query) and applied our rule-based model on top of each retrieved result, so that each web snippet is year-qualified. Then, in order to avoid biased results caused by possible incorrect temporal patterns, we manually checked whether each temporal expression was a correct date or not. Finally, we assess how strong each query is temporally related. To this end, we define three basic measures.

The first measure is $TTitle(q)$. It is defined in Equation 3.1 and can be seen as the ratio between the number of titles returned with years divided by the total number of titles retrieved for the query q. The other two measures are $TSnippet(q)$ and $TUrl(q)$ which are computed similarly for the snippet (descriptive text of the web snippet) and the URL. Both are respectively defined in Equations 3.2 and 3.3.

$$TTitle(q) = \frac{\#\ Year{-}Qualified\ Titles}{\#\ Titles\ Retrieved}, \tag{3.1}$$

$$TSnippet(q) = \frac{\#\ Year{-}Qualified\ Snippets}{\#\ Snippets\ Retrieved}, \tag{3.2}$$

$$TUrl(q) = \frac{\#\ Year{-}Qualified\ URLs}{\#\ URLs\ Retrieved}. \tag{3.3}$$

The average for all the queries is then determined by applying a micro-average scheme. The number of corresponding items returned for a query is added cumulatively to the values calculated for all the previously computed queries. $TTitle(.)$, $TSnippet(.)$ and $TUrl(.)$ are respectively defined in Equation 3.4, 3.5 and 3.6:

$$TTitle(.) = \frac{\sum_{i=1}^{|q|} \#Year{-}Qualified\ Titles_i}{\sum_{i=1}^{|q|} \#\ Titles\ Retrieved_i}, \quad (3.4)$$

$$TSnippet(.) = \frac{\sum_{i=1}^{|q|} \#Year{-}Qualified\ Snippets_i}{\sum_{i=1}^{|q|} \#\ Snippets\ Retrieved_i}, \quad (3.5)$$

$$TUrl(.) = \frac{\sum_{i=1}^{|q|} \#Year{-}Qualified\ URLs_i}{\sum_{i=1}^{|q|} \#\ URLs\ Retrieved_i}, \quad (3.6)$$

where |q| is the total number of queries.

The obtained results are shown in Table 3.1 and are according to our expectations.

Table 3.1: Average measure results in the Q465R20, Q450R20 and Q450R100 dataset.

Collections	**# Web Snippets Retrieved**	**# Items with Dates**		**Average Measures**		**# Dates Retrieved**	**# of ≠ Dates Retrieved**	**% Items with more than one Date**
Q465R20	16,648	Title	947	$TTitle(.)$	5.69%	1071	61	0.73%
		Snippet	2078	$TSnippet(.)$	12.4%	2916	161	3.74%
		Url	710	$TUrl(.)$	4.26%	643	48	0.26%
Q450R20	16,129	Title	481	$TTitle(.)$	2.98%	528	51	0.29%
		Snippet	1532	$TSnippet(.)$	9.50%	2048	161	2.46%
		Url	305	$TUrl(.)$	1.89%	327	45	0.17%
Q450R100	62,842	Title	2058	$TTitle(.)$	3.27%	2245	99	0.27%
		Snippet	5777	$TSnippet(.)$	9.19%	7486	220	2.28%
		Url	3512	$TUrl(.)$	5.59%	3738	100	0.41%

On average, about 9% of the snippets retrieved for the Q450R20 and Q450R100 collections have a temporal feature, i.e., 9.50% and 9.19% respectively. This contrasts with the 1.21% explicit temporal value of query logs previously determined in Section 2.3, which corroborates hypothesis **H1**: "*Web documents incorporate a high level of temporal information compared to available web query logs*".

This value is even significantly higher for the Q465R20 collection, as it includes 15 explicit temporal queries (e.g. "*hairstyles 2010*"), which naturally implies the retrieval of a larger range of correct outcomes. The occurrence of temporal features is particularly evident in the case of snippets, but still significant in the case of titles and URLs.

Another important issue is that the differences between Q450R20 and Q450R100 collections are minimal. The only exception comes from $TUrl(.)$, where the retrieval of a large number of results represents an increase from 1.89% to 5.59%. This is mainly due to the fact that early (top n) retrieved results are usually dynamic web pages with complex parameterized structures, while later results (tail n) are embodied by static links with well defined structures.

Moreover, although there is no noticeable difference between defining the retrieval of 20 or 100 results, with the abovementioned exception, getting more results will lead to the retrieval of a larger range of different dates. This may be certainly useful for a full understanding of the temporal references related to the query. As such, we will focus exclusively on the largest dataset in the remainder of this chapter.

Furthermore, we should call attention to the fact that dates often occur more than once in the same item. This is particularly evident in the case of snippets, with a value of 2.28%. This value can be better understood if instead of considering a relative measure where all the snippets are considered, we follow an absolute approach, where only those snippets having dates are taken into account. In that case, the values rises to approximately 23%.

Unsurprisingly, the occurrence of dates turns out to be much higher in recent years. This is clearly depicted in Figure 3.1, which denotes a trend for the emergence of dates from "*2003*" onwards, with particularly emphasis on the period of "*2008-2010*", which is not surprising given that this experiment was carried out in 2010.

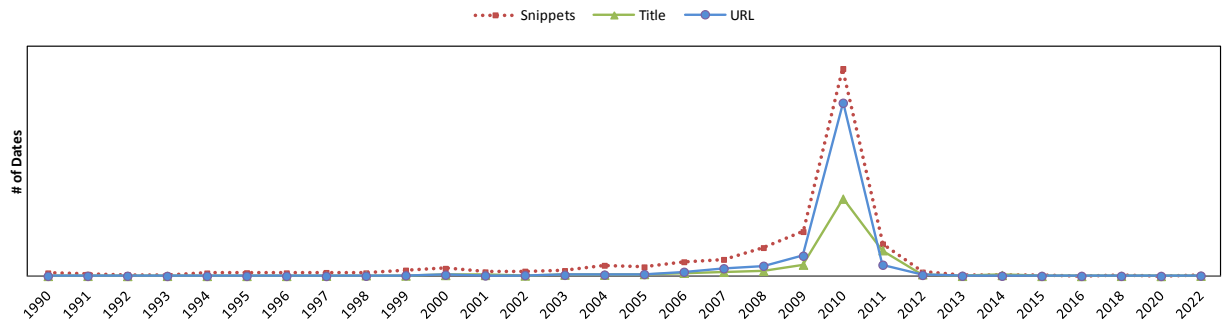


Figure 3.1: Distribution of dates from the Q450R100 dataset.

From Figure 3.2, we can also conclude that, irrespective of the item considered (titles, snippets or URLs), dates occur more frequently in response to queries belonging to the categories of Dates (e.g. *calendar*), Sports (e.g. *football*), Automotive (e.g. *dacia duster*), Society (e.g. *baby*) and Politics (e.g. *Barack Obama*).

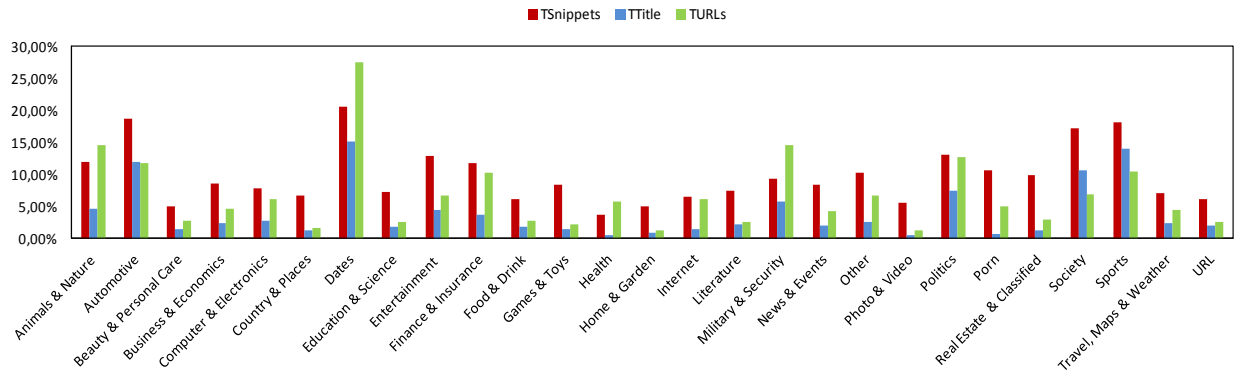


Figure 3.2: Distribution of dates per category from the Q450R100 dataset.

A more in-depth analysis shows how this information can be used to improve query understanding. For this purpose, we explore the results of two queries, "*tour de France*" and "*toyota recall*". For the first one, we rely on the $TSnippet(.)$ measure. We obtain a value of 77.78%, which clearly shows the temporality of the query. For the second query, we explore the positioning of dates in the timeline (see Figure 3.3). We show that, despite the occurrence of occasional temporal references over-time, the query has a clear evident break in "*2011*". This should be of interest to the user. In fact, it is related to Toyota's recall problem with the Prius model.

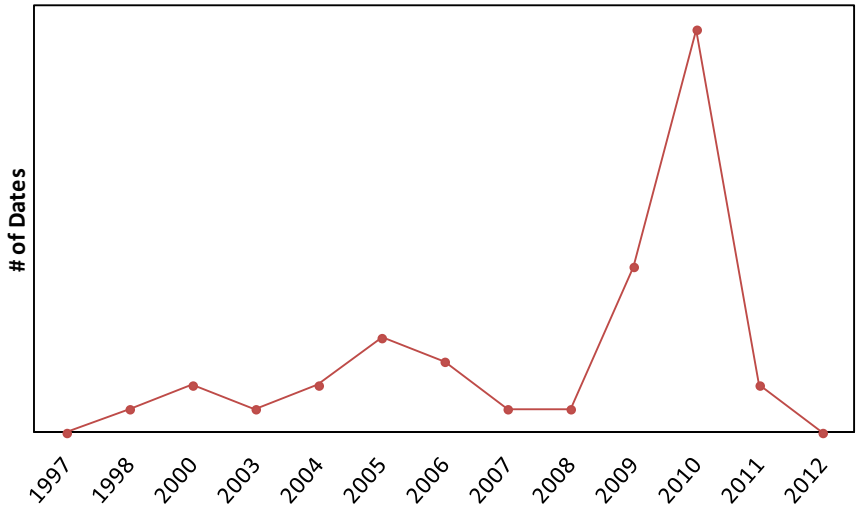


Figure 3.3: "*toyota recall*" query timeline for the "*1998-2011*" time span. Q450R100 dataset.

Finally, we measure the correlation between each of the three dimensions, $TTitle(.)$, $TSnippet(.)$ and $TUrl(.)$ in order to check whether these items behave similarly. With this goal in mind, we use the Pearson correlation coefficient [72]. The results indicate the strongest

correlation of 0.83 between the occurrence of dates in titles and snippets as clearly depicted in Figure 3.4.

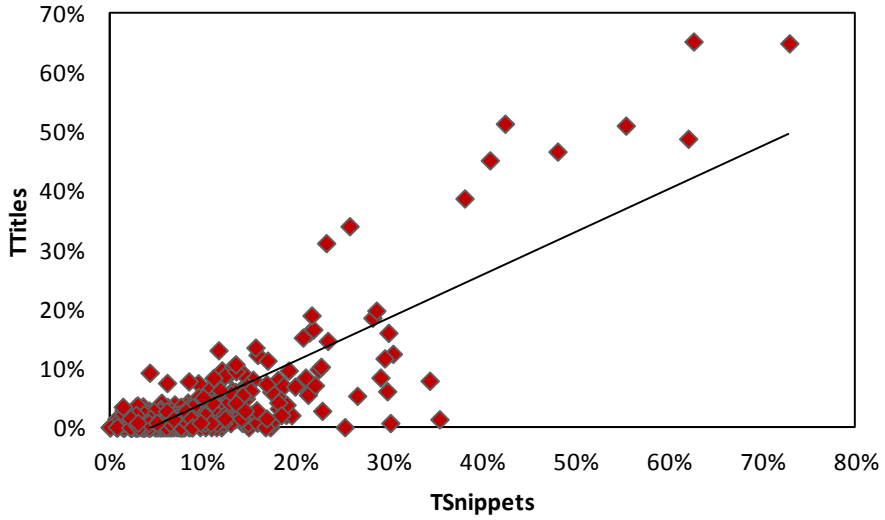


Figure 3.4: Snippet vs.. Title scatter plotter.

The provided evidence supports the claim that snippets contain a high level of temporal information that can be very useful in the process of dating implicit temporal queries and that this information is particularly linked to the occurrence of temporal information in text titles as well as in URLs in a smaller proportion. Moreover, the fact that URLs include some degree of noise, as previously shown in Table 2.2 led us not to consider this type of source. As a result, we simply rely on the extraction of temporal features within snippets and titles on the rest of our work. A summary of the overall results is given in Table 3.2 for the three different metrics.

Table 3.2: Pearson correlation between TTitle, TSnippet and TUrl.

Pearson Correlation	TTitle	TSnippet	TUrl
TTitle		0.83	0.69
TSnippet	0.83		0.57
TUrl	0.69	0.57	

3.2 Implicit Temporal Query Classification

We will now attempt to determine the prevalence of queries having a temporal nature. In order to achieve this, we rely on the set of 450 text queries that compose the Q450R100 collection and define two temporal classes in line with the work of Jones & Diaz [51]: *ATemporal*, i.e. queries

not sensitive to time (e.g. "*rabbit*"); *Temporal*, i.e. queries that either take place in a very concrete time period, known as temporally unambiguous (e.g. "*bp oil spill*") or that have multiple instances over time, known as temporally ambiguous (either occurring in a *periodic* fashion – e.g. "*SIGIR*" - or in an uncertainty aperiodical manner – e.g. "*oil spill*").

A preliminary step is required. Given that each query can have multiple meanings or facets, each one with different possible temporal dimensions, we need to first classify the query with regard to its conceptual nature, in a way that only single meanings or facets can be given a temporal tag. For this first step, we follow the approach of Song et al. [82], who define three types of concept queries: ambiguous, broad and clear, all described in Table 3.3.

Table 3.3: Query concept classification. Adapted from Song et al. [82].

Type of Query	Description	Example
Ambiguous	A query that has more than one meaning.	"*scorpions*", which may either refer to the *rock band*, the *arachnid* or the *zodiac sign*.
Broad	A query that covers a variety of subtopics.	"*quotes*", which covers some subtopics such as *love quotes*, *historical quotes*, etc.
Clear	A query that has a specific meaning and covers a narrow topic. Usually is a successful search in which the user can find what he is looking for in the first page of results.	*"bank of America*"

For the purpose of query concept classification, we used the disambiguation Wikipedia feature, which helps to understand whether a query has more than one meaning. The remaining queries are either classified as broad or clear, depending on whether they have more than one facet or not. For that purpose, we used the HISGK-means ephemeral clustering algorithm [35] and based on the discovered clusters, a human judge decided upon classification. Final results for the Q450R100 dataset (see Table 3.4) show that most of the queries are ambiguous in concept, followed very closely by clear queries. Broad queries on the other hand are just a simple fraction.

Table 3.4: Concept query classification of the Q450R100 dataset.

Conceptual Classification	Number of Queries
Ambiguous	220
Broad	54
Clear	176

Each clear concept query must then be classified into one of the two temporal classes mentioned above, i.e., Temporal and ATemporal. For this purpose, we defined a simple Temporal

Ambiguity query function, denoted $TA(q)$, that linearly combines TTitle, TSnippet and TUrl to determine the aggregated temporal value of the query. $TA(q)$ is defined in equation 3.7:

$$TA(q) = \sum_{f \in I} \omega_f \,.\, f(q), I = \{TTitle, TSnippet, TUrl\}, \quad (3.7)$$

where ω_f is the weight of the I measures and $f(q)$ is the corresponding value obtained for the query q. Since we rely on the extraction of temporal features only within Titles and Snippets, we consider a value of 0 for ω_{TUrl}.

Moreover, instead of considering a value of 50% for ω_f, both for TTitle and TSnippet, we defined a weighted average that gives more importance to the *I* item that incorporates the highest number of temporal features possible. As such, given that $TTitle(.)$ equals 3.27% and $TSnippet(.)$ is 9.19% (recall Table 3.1) we set ω_f as 26.27% for TTitle and 73.73% for TSnippet. Table 3.5 shows an example of the computation of $TA(q)$ for three different queries, "*twilight eclipse*", "*toyota recall*" and "*hdf netbanking*".

Table 3.5: Temporal ambiguity for the queries "*twilight eclipse*", "*toyota recall*", "*hdf netbanking*".

Query	$TTitle(q)$	$TSnippet(q)$	$TA(q)$
twilight eclipse	6.8%	14.3%	12.3%
toyota recall	16.5%	21.8%	20.4%
hdf netbanking	0.0%	2.9%	2.1%

A query is then defined to be *ATemporal* if its $TA(q)$ is below a given θ value and as $Temporal$ otherwise. The classification function is defined in Equation 3.8:

$$TQC(q) = \begin{cases} TA(q) > \theta, Temporal \\ TA(q) \leq \theta, ATemporal \end{cases}. \quad (3.8)$$

In order to evaluate our simple classification model, we asked three human annotators to judge the set of 176 clear concept queries with regard to their temporality. Human annotators were asked to consider each query, to look at web search results and to classify them as $Temporal$ or $ATemporal$. The final classification of each query comes by majority voting. As such, each query is considered to be ATemporal if it gets at least two votes, while Temporal

otherwise. Overall results pointed at 26.7% of implicit temporal queries, whereas 73.3% of atemporal ones (see Table 3.6).

Table 3.6: Manual temporal query classification of the Q450R100 dataset.

Temporal Classification	Number Queries	%
ATemporal	129	73.3%
Temporal	47	26.7%

An inter-rater reliability analysis using the Fleiss Kappa statistics [42] was then performed to determine consistency among annotators. Results have shown a value of 0.89, thus indicating an almost perfect agreement between the raters.

The same statistic test was then used to determine the consistency among the $TQC(q)$ function and the three human annotators over different values of θ. The obtained results are depicted in Figure 3.5 and show that Fleiss Kappa is maximized for $\theta = 0.11$ with an overall test value of 0.71.

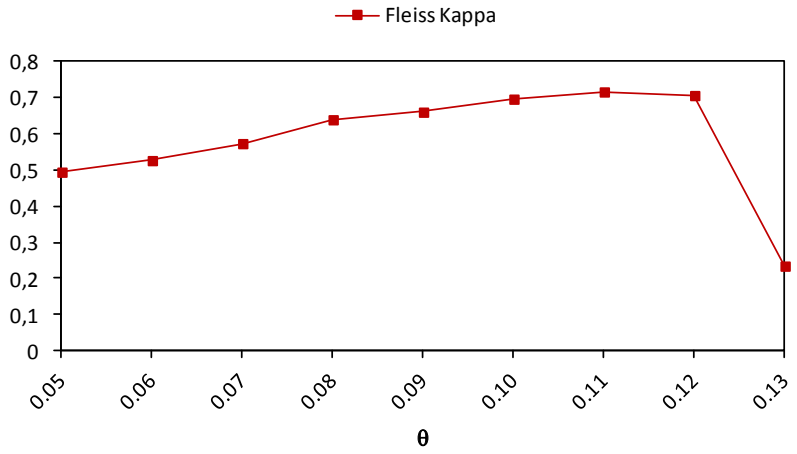


Figure 3.5: Fleiss Kappa values when varying θ for the $TQC(q)$ function .

This experiment was then complemented with a further evaluation so as to make sure that θ was correctly determined. For this purpose, we compared the results of the human annotators majority voting final classification with the results that stem from applying the $TA(q)$ function for each query q. The best θ is then determined by applying a classical IR evaluation supported on the calculation of *Precision* (Equation 3.9), *Recall* or *Sensitivity* (Equation 3.10), *F*1-*Measure* (Equation 3.11) and *Balanced Accuracy* or *Efficiency* (Equation 3.12):

$$Precision\ (P) = \frac{TP}{TP+FP}, \tag{3.9}$$

$$Recall\ (R) = \frac{TP}{TP+FN}, \tag{3.10}$$

$$F1 - Measure\ (F1 - M) = \frac{2*Precision*Recall}{Precision+Recall}, \tag{3.11}$$

$$Balanced\ Accuracy\ (BA) = \frac{0.5*TP}{TP+FN} + \frac{0.5*TN}{TN+FP}, \tag{3.12}$$

with True Positives (*TP*) being the number of queries correctly identified by the TQC function as Temporal, True Negatives (*TN*) being the number of queries correctly identified by the TQC function as ATemporal, False Positive (*FP*) being the number of queries wrongly identified by the TQC function as Temporal and False Negative (*FN*) being the number of queries wrongly identified by the TQC function as ATemporal. A representation of this is given in Table 3.7.

Table 3.7: Confusion matrix representation.

		Gold Standard (Human Annotators)	
		Positive	**Negative**
Outcome (*TQC*)	**Positive**	TP	FP
	Negative	FN	TN

In order to avoid over-fitting and understand the generalization of the results, we followed a *n*-fold repeated random sub-sampling approach. This method randomly splits the dataset into disjoint training and test sets *n* times. For each partition, the model is fit in the training set at some fixed ratio (usually, ~80% of the observations), and performance is estimated by applying the resulting classification to the testing examples (~20%). For the computation of the IR metrics we applied a micro-average approach where TP, FP, FN and TN are first summed up before being computed. Final results for the set of all test datasets are then determined by averaging the accuracies of the *n* individual folds. More specifically, we used stratified *5*-fold with 80% of learning instances for training and 20% for testing.

The obtained results (see Table 3.8) show that $TQC(q)$ is capable of achieving 69.4% F1 performance, 77.7% of Balanced Accuracy (*BA*), 79.1% of Precision (*P*) and 62.6% of Recall (*R*) matching a cutoff of θ = 0.11, which is in line with the θ value determined by the Fleiss Kappa test.

Table 3.8: Stratified *5*-fold repeated random sub-sampling test dataset for the $TQC(q)$ function.

Test Dataset (20%)	Training Cutoff	TP	TN	FP	FN	R	BA	P	F1-M
D1	0.11	8	22	3	3	0.727	0.804	0.727	0.727
D2	0.11	7	27	0	2	0.778	0.889	1.000	0.875
D3	0.11	5	24	3	4	0.556	0.722	0.625	0.588
D4	0.11	8	22	1	5	0.615	0.786	0.889	0.727
D5	0.11	5	23	2	6	0.455	0.687	0.714	0.556
Average	0.11	-	-	-	-	0.626	0.777	0.791	0.694

We then apply θ to automatically classify each of the 176 clear concept queries with regard to its temporality. The classification results (see Table 3.9) show that of all the clear concept queries, 22% have an implicit temporal nature and that 78% of them are ATemporal queries. These values contrast with those presented by Metzler et al. [65] who, based on web query logs, estimated that only 7% of the queries have an implicit temporal nature. This gives more strength to hypothesis **H2** which states that "*There is a significant difference between temporally classifying a query based on information extracted from the contents of the web documents or from web query logs*".

Table 3.9: Automatic temporal query classification of the Q450R100 dataset.

Temporal Classification	Number Queries	%
ATemporal	137	78%
Temporal	39	22%

Likewise, these values contrast with the results obtained from our human annotators task, which pointed at 26.7% of implicit temporal queries from human annotators, while only 22% were given by our methodology. A detailed analysis of the results presented in Table 3.8 shows that this difference is mostly due to some False Negative classifications causing $TQC(q)$ not to retrieve some real temporal queries. On these grounds, we can conclude that the temporal information found within web snippets is not enough to correctly classify some of the queries with regard to their temporality. One possible solution is to complement the Temporal Ambiguity query function with further temporal information. This should be addressed in the future.

In next part, we shall compare the temporal value of web snippets to the temporal value of web query logs.

3.3 Comparing the Temporal Value of Web Snippets with Web Query Logs

In this section, we aim to quantify the temporal value of Yahoo! and Google query logs accessible through their respective completion engines and compare it to web snippets. To pursue this, we rely again on the GISQC_DS dataset and on the set of 176 clear concept queries selected from the Q450R100 collection and introduce two measures, called $TLogYahoo(q)$ and $TLogGoogle(q)$, in a similar way as $TSnippet(q)$ but in the context of web usage. $TLogYahoo(q)$ and $TLogGoogle(q)$ are defined in Equation 3.9 and 3.10 respectively as the ratio between the number of suggested queries associated with years divided by the total number of retrieved queries from the completion engine, which is 10:

$$TLogYahoo(q) = \frac{\#\ Year{-}qualified\ Queries\ Retrieved\ from\ Yahoo}{\#\ Queries\ Retrieved\ from\ Yahoo}, \tag{3.13}$$

$$TLogGoogle(q) = \frac{\#\ Year{-}qualified\ \ Queries\ Retrieved\ from\ Google}{\#\ Queries\ Retrieved\ from\ Google}. \tag{3.14}$$

In order to understand better the computation of these values, we present an example for the query "*bp oil spill*". We divide Figure 3.6 into two parts: the left hand side concerns the results of Yahoo! and the right one Google results. Among all the results, only a single date ("*2010*") is found, in particular within the Yahoo! search engine. As a result $TLogYahoo(Bp\ Oil\ Spill)$ would equal to 0.1, while $TLogGoogle(bp\ oil\ spill)$ would be equal to 0.

Figure 3.6: Yahoo! and Google query suggestion for the query "*bp oil spill*".

After having computed these values for all the queries, we then compare the temporal value of query logs with the temporal value of web snippets. For this purpose, we calculated the Pearson correlation coefficient between $TLogGoogle(q)$, $TLogYahoo(q)$, $TSnippet(q)$, $TTitle(q)$ and $TUrl(q)$. Final results (see Table 3.10) show that the best correlation values occur between $TTitle(q)$ and $TLogGoogle(q)$ with a value of 0.69 and between $TSnippet(q)$ and $TLogGoogle(q)$ with 0.63.

Table 3.10: Pearson correlation coefficient between TLogYahoo, TLogGoogle, TTitle, TSnippet, and TUrl.

Pearson Correlation	TLogGoogle	TTitle	TSnippet	TUrl
TLogYahoo	0.63	0.61	0.52	0.48
TLogGoogle		0.69	0.63	0.44

These results are complemented with two scatter plots. An overall analysis of Figure 3.7 and Figure 3.8 shows that while most of the queries have a $TSnippet(q)$ values around 20%, $TLogYahoo(q)$ and $TLogGoogle(q)$ are mostly near to 0%.

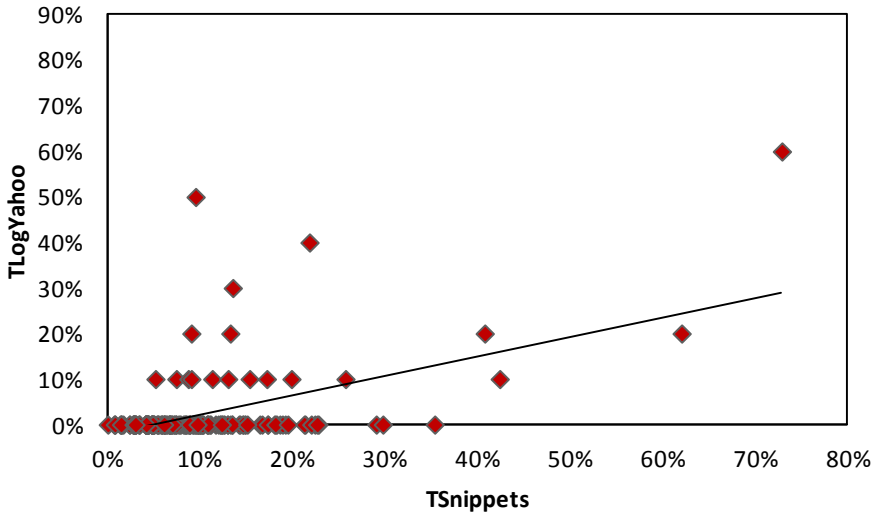


Figure 3.7: $Tsnippet(q)$ vs.. $TLogYahoo(q)$ scatter plot.

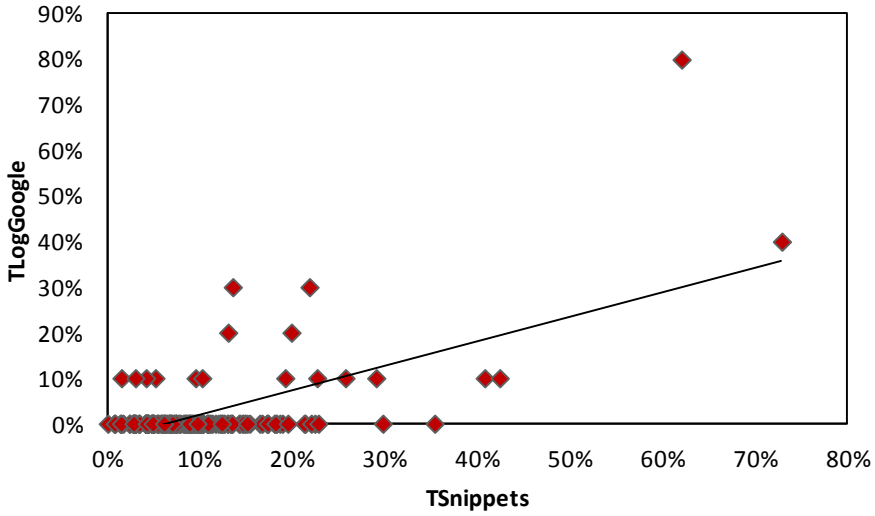


Figure 3.8: $Tsnippets(q)$ vs. $TLogGoogle(q)$ scatter plot.

Finally, we determine which of the two collections (web snippets or web query logs) retrieves a wider range of different dates[22]. In order to achieve that, we built a confidence interval for the difference of means, for paired samples, between the number of different years appearing in the web snippets retrieved for a given query and in the web query logs for the same query. The intervals obtained, [5.10; 6.38] for $TLogYahoo(.)$ and [5.12; 6.43] for $TLogGoogle(.)$, show, with 95% confidence, that the number of different years appearing in web snippets is significantly higher than in either one of the two web query logs, e.g. at the minimum of the interval, there are on average five times more different years in web snippets than in web query logs. We can conclude that web snippets present a higher diversity of dates, which gives more strength to hypothesis **H1**: "*Web documents incorporate a high level of temporal information compared to available web query logs*".

3.4 Summary

In this chapter, we sought to study the temporal value of web snippets and to discuss the extent to which the temporal information found can be used to classify implicit temporal queries. The experiments conducted over one dataset made publicly available (GISQC_DS) has shown that on average, about 9% of the snippets retrieved for a text query have dates. This contrasts with the 1.21% explicit temporal value of query logs previously determined in Section 2.3. A further experiment has shown that web snippets also retrieve a larger range of different dates. This constitutes evidence that web snippets are a very useful data source that can help in the process of classifying implicit temporal queries. In this regard, we showed that 22% to 26.7% of the queries classified upon information extracted from the corresponding web snippets have an implicit temporal nature. This clearly contrasts with the work of Metzler et al. [65] who, based on web query logs, determined that only 7% of queries have an implicit temporal nature. Overall, we can draw the conclusion that web snippets present a large temporal value, which can be very useful to infer the query's temporal nature. We will pursue this research direction in the coming chapter.

[22] Note that while in the case of web snippets we may potentially consider a maximum of 100 retrieved results, in the case of the completion engines we are forced to simple look at a maximum of 10 results per query.

Chapter 4

Temporal Disambiguation of Queries

To understand the temporal intent of a query formulated by a user is a particularly hard task. While in the case of explicit temporal queries (e.g. "*Fukushima 2011*") the retrieval task can be relatively straightforward, in the case of implicit temporal ones (e.g. "*Iraq war*") it is much more complex since it involves estimating the temporal part of the query. Given that most of the temporal queries issued by users are implicit by nature (as shown in the previous chapter), grasping its underlying temporal intent is a significant challenge and a necessary condition of any improvement in the performance of search systems. In this context, most state-of-the-art methodologies rely on existing temporal annotation tools, considering any occurrence of temporal expressions in web snippets and other web data, as equally relevant to an implicit temporal query. However, applying time-taggers to web collections based on simple regular expressions is likely to have a negative impact on system effectiveness. As noted previously, this is mainly due to the mere fact that the simple identification of a year pattern may not be enough to determine whether it is a real date or it is relevant to the query. An enlightening example is given for the query "*Haiti earthquake*", which may retrieve the following web snippet.

> *2011* Haiti Earthquake Anniversary
> As of *2010* (see *1500* photos), the following major earthquakes have been recorded in Haiti. The 1st one occurred in *1564*. *2010* has been a tragic date, however in *2012* Haiti will organize the Carnival…

Example 4.1: Web snippet temporal information extraction for the query "*Haiti earthquake*".

While there are a few year candidates, only "*1564"* and "*2010"* are relevant to the query. "*2012"* is not query-related, "*1500"* is not even a date and "*2011"* may be considered relevant for the anniversary facet and not for the event itself.

If we can automatically identify this information, we are then able to improve the overall performance of several T-IR tasks. This may be potentially useful, for example, for temporal query understanding, temporal ranking of documents or temporal clustering.

In this chapter[23], we propose a language-independent strategy that associates top relevant years to any text query, while filtering out non-relevant ones. Since results are produced "on-the-fly", we adopt a web content analysis approach over the set of *n*-top web snippets retrieved in response to the user's query. This contrasts with an analysis of full web pages, which requires a more complex infrastructure, which is out of the scope of this thesis. In order to accomplish our objectives we adopt a two-folded approach.

1. Firstly, we present our Generic Temporal Evaluation measure (*GTE*), which evaluates the temporal similarity between a query and a candidate date;
2. Secondly, we propose a classification model (*GTE-Class*) so to accurately relate relevant dates to their corresponding query terms and filter out non-relevant ones. With respect to this, we suggest two different solutions:
    - A threshold-based classification strategy;
    - A supervised classifier based on a combination of multiple similarity measures.

We finally evaluate both strategies over a set of real-world text queries and compare the performance of our web snippet approach with a query log one, over the same set of queries.

Our contributions in this chapter can be summarized as follows: (1) we propose a novel approach to tag text queries with relevant temporal expressions by relying on a content-based approach and a language-independent methodology; (2) our generic temporal similarity measure, GTE, outperforms well-known first order similarity measures, including web-based ones; (3) our proposal improves precision in a task of date tagging with respect to a query-log based approach; (4) we make available to the scientific community a set of queries and ground-truth results, fostering the development and relative assessment of future approaches and (5) we provide a few

[23] This chapter is partially based on the work published at the 2nd International Temporal Web Analytics Workshop associated with WWW2012 (Campos et al. 2012a) and the 21st ACM International Conference on Information and Knowledge Management - CIKM 2012 (Campos et al. 2012b).

web services to both the scientific community and the general public so that GTE and GTE-Class can be tested and visible to a larger audience.

This chapter is structured in 5 sections. Section 4.1 offers an overview of related research. Section 4.2 defines both the GTE and the classification methodology (GTE-Class). Section 4.3 describes the experimental setup. Section 4.4 discusses the obtained results. Finally, Section 4.5 summarizes the chapter and ends with some final remarks.

4.1 Related Research

Within the overall context of T-IR, Jones & Diaz [51] were the first to consider implicit temporal queries. In their work, the authors follow a metadata-based approach, using a language model solution and a collection of web news documents to model the period of time that is relevant to a query. Dakka et al. [31] estimate the important times of the query by analyzing the number of documents matching the query over time (based on the publication time of the document) to subsequently incorporate time into language models. Kanhabua & Nørvåg [53], on the other hand, propose three different methods to determine the time of queries. They rely on the use of temporal language models, based on a New York Times (*NYT*) news collection, where documents are explicitly time-stamped with the document creation time. Finally, Matthews et al. [63], combine a metadata-based and a content based approach to analyze how NYT news topics change over time. Unfortunately, all of these approaches are language-dependent and mainly rely on the creation date of the documents as the correct temporal issue, which is far from being true in most of cases. Moreover, such information is not even available in the majority of the documents and not even available in the majority of the documents.

An alternative solution to using metadata is proposed by Metzler et al. [65] who suggest to mine query logs in order to identify implicit temporal information needs. In their work, the authors propose a weighted measure that considers the number of times a query, q, is pre and post-qualified with a given year, y. A query is then implicitly year qualified if it is qualified by at least two different years. A relevance value is then given for each year found in the document. Based on this, the authors propose a time-dependent ranking model that explicitly adjusts the score of a document in favor of those matching the users' implicit temporal intents. The referred study proposes an interesting solution as it introduces the notion of correlation between a query and a year. However, the approach lacks in query coverage as it depends on query logs analysis.

A third possibility for dating implicit temporal queries is to consider temporal information extracted from web contents. To the best of our knowledge only one enquiry [56] has taken up this research so far, however in the context of document relevance rather than that of query relevance. More specifically, Kawai et al. [56], developed a chronological events search engine for the Japanese language based on web snippets analysis. In order to collect a large number of temporal expressions, the authors expand the query with language dependent expressions related to event information such as past and future year expressions, temporal modifiers and context terms. Then, noisy temporal patterns are removed from sentences using machine learning techniques trained over a set of text features. While the incorporation of a date filtering process is a novelty, considering a content approach, this study does yet not determine a degree of relevance for each temporal pattern found.

Such an approach was first addressed by Strötgen et al. [85] in the context of document relevance. More specifically, the authors propose an enriched temporal profile for each document, where each temporal expression found is represented by a larger number of different features. Final relevance then emerges from the combination of all the features into a single relevance function based on a set of pre-defined heuristics. However, this study lacks a further evaluation in terms of IR metrics.

Our approach differs from previous research on dating queries in several aspects. Firstly, we do not make use of query logs or metadata information. Moreover, we do not resort to a set of heuristics extracted from a document's content or a supervised classification methodology. Instead, in our approach, we detect relevant temporal expressions based on corpus statistics and a general similarity measure that makes use of co-occurrences of words and years extracted from the contents of the web snippets. Secondly, our methodology is language-independent as we do not use any linguistic-based techniques. Instead, we use a rule-based model solution supported by language-independent regular expressions. Finally, apart from estimating the degree of relevance of a temporal expression, we present an appropriate classification strategy to determine whether or not a date is query relevant. This is the first main contribution of this thesis.

4.2 Identifying Query Relevant Temporal Expressions

In this section, we describe the method that guides our identification of top relevant dates related to text queries with a temporal dimension. We rely on the extraction of temporal information from

the text itself, particularly within the set of *n*-top web snippets returned in response to a query. As shown by Alonso et al. [3, 6], this type of collection is an interesting alternative for the representation of web documents, where years often appear as discussed in the previous chapter. Although we have focused on web snippets in our experiments, our temporal similarity measure is equally applicable to any document collection embodying temporal information, such as Wikipedia pages or Twitter posts.

The overall idea of the process is to identify and classify years which are relevant for a given query on four different steps depicted in Figure 4.1 and explained in the remainder of this section: web search, web snippet representation, temporal similarity and date filtering. In particular, this will build the foundations for the two applications developed in this thesis: temporal clustering and temporal re-ranking.

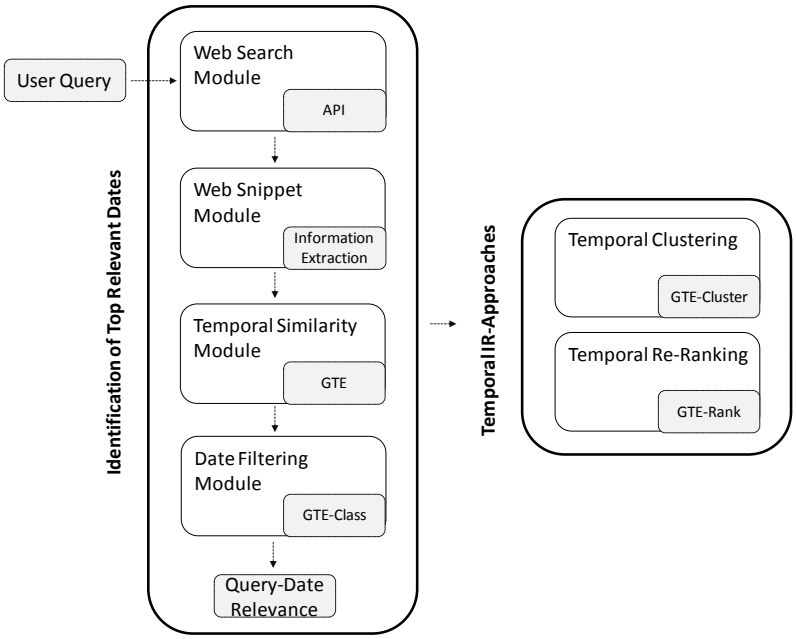


Figure 4.1: GTE overall architecture.

4.2.1 Web Search

We assume a query to be either explicit, i.e., a combination of both text and time, denoted q_{time}, or implicit, i.e. just text, denoted q_{text}. In this thesis, we deal with the latter since handling explicit temporal queries is a less complex task. For the sake of readability, we denote a query simply as q. Similarly to Kawai et al. [56], we use a prospective search where the query is first issued before results are gathered and indexed. For the purposes of collecting the results, we use a web search API to access an up-to-date index search engine. Given a text query q, we obtain, as the result of the search, a collection of n web snippets $S = \{S_1, S_2, \ldots, S_n\}$.

4.2.2 Web Snippet Representation

Each S_i, for $i = 1, \dots, n$, denotes the concatenation of two texts, i.e. $\{Title_i, Snippet_i\}$ and is represented by a bag-of-relevant-words and a set of candidate temporal expressions. In what follows, we assume that each S_i is composed by two different sets denoted W_{S_i} and D_{S_i}:

$$S_i \rightarrow \left(W_{S_i}, D_{S_i}\right), \tag{4.1}$$

where $W_{S_i} = \{w_{1,i}, w_{2,i}, \dots, w_{k,i}\}$ is the set of the k most relevant words/multiwords associated with a web snippet S_i and $D_{S_i} = \{d_{1,i}, d_{2,i}, \dots, d_{t,i}\}$ is the set of the t candidate years associated with a web snippet S_i. Moreover,

$$W_S = \bigcup_{i=1}^{n} W_{S_i}\ , \tag{4.2}$$

is the set of distinct relevant words/multiwords (hereafter called words) extracted for a query q, within the set of web snippets S, i.e. the relevant vocabulary. In this thesis, relevant words are identified using a web service[24] provided by Machado et al. [59, 60], which selects words and multiwords based on a specific segmentation process and a numeric selection heuristic.

Similarly,

$$D_S = \bigcup_{i=1}^{n} D_{S_i}\ , \tag{4.3}$$

is defined as the set of distinct candidate years extracted from the set of all web snippets S. For this purpose, a simple rule-based model, as introduced in Section 2.3, is used to extract explicit temporal patterns.

Finally,

$$W^* = W_S \cap W_{d_j}\ , \tag{4.4}$$

is defined as the set of distinct words that results from the intersection between the set of words W_S and the set W_{d_j} which contains the distinct words W_h that appear together with the candidate date d_j in every web snippet S_i of S, as explained hereafter:

$$W_{d_j} = \left\{W_h : W_h \in \left(W_{S_i}, .\right) \wedge\ d_j\ \in \left(., D_{S_i}\right), \forall\ S_i\right\}. \tag{4.5}$$

[24] http://wia.info.unicaen.fr/TokenExtractor/api/Token?query= [February 25th, 2013]

To illustrate our approach we present a running example for the query "*Haiti earthquake*". Table 4.1 lists the set of three web snippets retrieved upon the query execution and the formed sets, W_{S_i} and D_{S_i}.

Table 4.1: Running Example: Haiti earthquake.

$Title_1$	**2011 Haiti Earthquake** Anniversary
$Snippet_1$	As of **2010** (see **1500** photos here), the following **major earthquakes** have been recorded in **Haiti**. The first one occurred in **1564**.
W_{S_1}	*haiti earthquake; major earthquakes; Haiti*
D_{S_1}	*1500; 1564; 2010; 2011*
$Title_2$	**Haiti Earthquake** Relief
$Snippet_2$	On January 12, **2010**, a massive earthquake struck the nation of **Haiti**, causing **catastrophic damage** inside and around the capital city of **Port-au-Prince**.
W_{S_2}	*haiti earthquake; haiti; catastrophic damage; Port-au-Prince*
D_{S_2}	*2010*
$Title_3$	**Haiti Earthquake**
$Snippet_3$	The first great earthquake mentioned in histories of **Haiti** occurred in **1564** in what was still the Spanish colony. It destroyed **Concepción de la Vega**.
W_{S_3}	*haiti earthquake; haiti; Concepción de la Vega*
D_{S_3}	*1564*

W_S and D_S are defined as two distinct sets, {*haiti earthquake; major earthquakes; haiti; catastrophic damage; Port-au-Prince; Concepción de la Vega*} and {*1500; 1564; 2010; 2011*} respectively. Each candidate date is then assessed with regard to its temporal similarity with the query. We formalize this process in the following section.

4.2.3 GTE: Temporal Similarity Measure

We formally define the problem of (query, candidate date) temporal relevance as follows: given a query q and a candidate date $d_j \in D_S$ assign a degree of relevance to each (q, d_j) pair. To model this relevance, we will use a temporal similarity measure, SIM, to be defined, ranging between 0 and 1:

$$SIM(q, d_j) \in [0,1]. \tag{4.6}$$

The aim is to identify dates d_j, which are relevant for q and minimize any errors caused by non-relevant or wrong dates. Our proposal is that the relevance between a (q, d_j) pair is better defined if, instead of just focusing on the self-similarity between the query q and the candidate date d_j, all the information existing between W^* and d_j is considered. Considering the candidate date *2010* of our running example, this means that we should take into account not only the similarity between *2010* and the query "*Haiti earthquake*", but also all the similarities occurred between *2010* and W^*, identified in Table 4.2 with an "X". Similarly, we should process all the similarities between *1500*, *1564*, *2011* and the corresponding W^*.

Table 4.2: List of words that co-occur with *2010*.

	W_S	2010
W^*	Haiti earthquake	X
	major earthquakes	X
	Haiti	X
	catastrophic damage	X
	Port-au-Prince	X
	Concepción de la Vega	--

Our assumption is based on the following principle:

P4.1: The more a given candidate date is correlated to the set of corresponding, distinct and most relevant words associated with the query - i.e. the intersection between the set of words relevant with the query, W_S, and the set of words W^* co-occurring with the candidate date - the more the query will be associated with the candidate date.

Thus, we will not only define the similarity between the query words q and the candidate date d_j, but also between each of the most important words $w_j \in W^*$ extracted from the set of web snippets and the respective candidate date d_j. Our proposal for the measure SIM is GTE, which is presented in Equation 4.7, where sim represents any similarity measure of first or second-order and F an aggregation function of the several $sim(w_j, d_j)$:

$$GTE(q, d_j) = F(sim(w_j, d_j)),\ w_j \in W^*. \tag{4.7}$$

We describe each of these two topics, sim and F, as follows.

sim similarity measure

In this thesis, sim represents a similarity measure, either of first or second order. While first order association measures evaluate the relatedness between two words as they co-occur in a given context (e.g. ngram, sentence, paragraph, corpus), second order co-occurrence measures are based on the principle that two words are similar if their corresponding context vectors are also similar. Here, we define a context vector as a set of tokens that co-occur somehow with the target word and the target candidate date. Figure 4.2 shows an example for both types of measures. In the figure, w_j represents one of the several possible words of W^*, for example *Port-au-Prince* and d_j one candidate date, for instance *2010*. Each empty box in turn represents one token of the corresponding context vector.

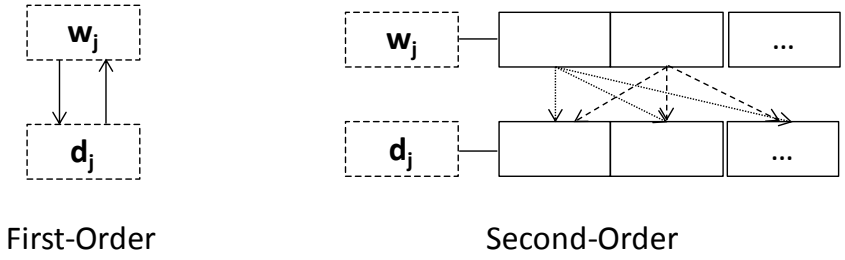


Figure 4.2: Example of first-order and second-order similarity measures.

Our hypothesis, which will be supported in the experiments section, is that second order similarity measures carry valuable additional relations in both the word w_j and the candidate date d_j context vectors, which cannot be induced if a direct co-occurrence approach between w_j and d_j is used. In this context, most of the works apply the cosine similarity measure. However, as most of them rely on exact matches of context words, their accuracy is low since language is creative and ambiguous [43]. This is particularly evident in the case of relations between words and temporal patterns, where the cosine similarity measure may not even be applied. In order to overcome these challenges, other measures have been proposed. More specifically, Ikehara et al. [44] proposed the semantic vector space model. Its basic idea was to calculate the Cosine coefficient between two word vectors, augmented with their concepts found in WordNet. However, one of the problems of this measure is that it suffers from language dependency. Deerwester et al. [32] on the other hand, proposed the Latent Semantic Analysis (*LSA*). Although LSA has shown interesting results in different areas [39], it has also shown inefficiency when compared to other similarity measures, as highlighted by Turney [86].

In this thesis, we apply the InfoSimba (*IS*) second-order similarity measure, a vector space model supported by corpus-based token correlations proposed by Dias et al. [33] as defined in Equation 4.8:

$$IS(w_j, d_j) = \frac{\sum_{i \in X} \sum_{j \in Y} S(i,j)}{\left(\begin{matrix} \sum_{i \in X} \sum_{j \in X} S(i,j) + \\ \sum_{i \in Y} \sum_{j \in Y} S(i,j) - \\ \sum_{i \in X} \sum_{j \in Y} S(i,j) \end{matrix} \right)}. \tag{4.8}$$

IS calculates the correlation between all pairs of two context vectors X and Y, where X is the context vector representation of w_j and Y is the context vector representation of d_j. To define the context vectors, we have at least five possible representations: (*W*;*W*), (*D*;*D*), (*W*;*D*), (*D*;*W*) and (*WD*;*WD*), where *W* stands for a word-only context vector, *D* for a date-only one and *WD* for a combination of words and dates. A clear picture of all the possible representations is given in Figure 4.3, where $(w_1, w_2, .., w_N)$ and $(d_1, d_2, .., d_N)$ are the elements of the two context vectors.

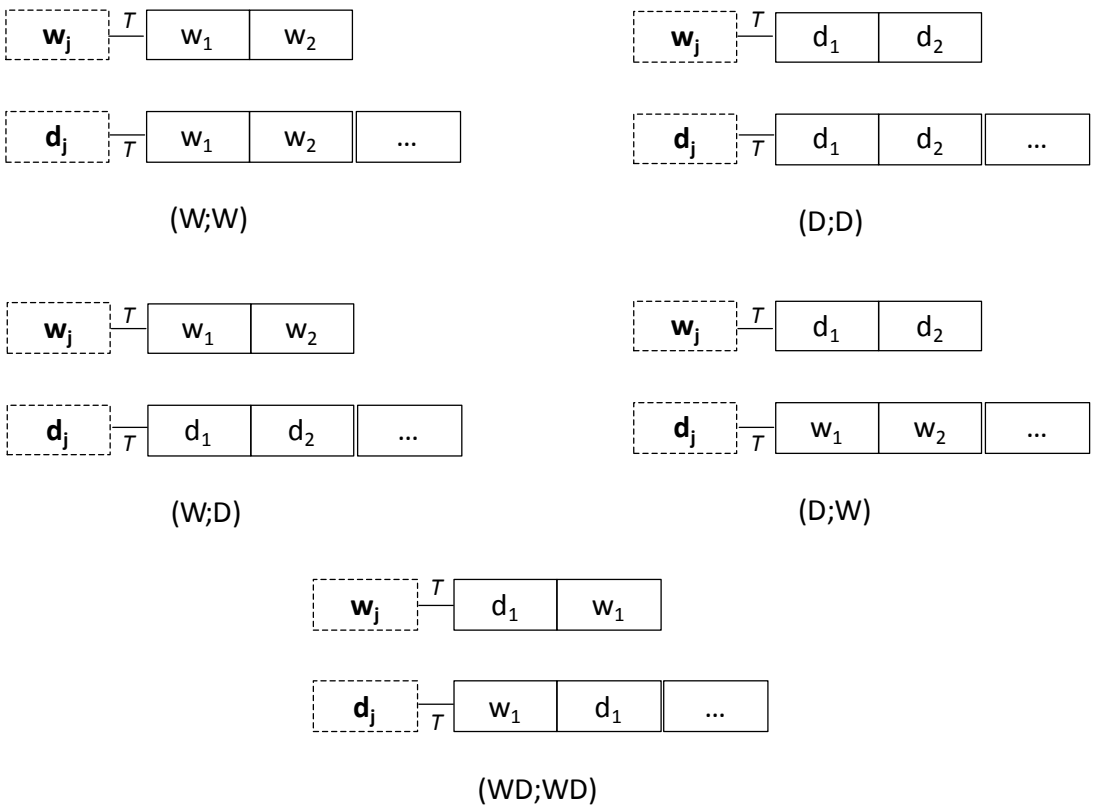


Figure 4.3: Context vector representations: (W;W); (D;D); (W;D); (D;W); (WD;WD).

Furthermore, we have to define the size of the context vector, denoted N and a threshold similarity value T. This threshold is the minimum similarity value above which, words and candidate dates should be selected as elements of the two context vectors. For that end, a conceptual temporal correlation matrix, which stores the similarity between the most important words and the candidate dates is built. M_{ct} is defined in Equation 4.9 as the "*word*"-"*word*", "*candidate date*"-"*candidate date*" and "*word*"-"*candidate date*" matrices respectively containing the normalized S similarities, where S is any first order similarity measure (e.g., *Pointwise Mutual Information*, *Symmetric Conditional Probability* or *DICE coefficient*):

$$M_{ct} = \begin{bmatrix} A_{k\times k} & B_{k\times t} \\ B^T_{t\times k} & C_{t\times t} \end{bmatrix}_{(k+t)\times(k+t)}, \tag{4.9}$$

where $A_{k\times k}$ is the $k \times k$ matrix which represents the similarity between k words, $C_{t\times t}$ is the $t \times t$ matrix which represents the similarity between t candidate dates, $B_{k\times t}$ is the $k \times t$ matrix which represents the similarity between k words and t candidate dates, and $B^T_{t\times k}$ is the transposition of the matrix.

To determine the context vector of a candidate date d_j for the representation type (WD;WD), with $T > 0$, only those words $(w_1, w_2, .., w_N)$ and candidate dates $(d_1, d_2, .., d_N)$ having a minimum S similarity value (>0) with $(., d_j)$[25] are eligible for the context vector. Likewise, S, would relate all the possible combinations $(w_j, .)$ that would enable us to determine the set of words $(w_1, w_2, .., w_N)$ and candidate dates $(d_1, d_2, .., d_N)$ that should be part of the w_j context vector.

We illustrate this in Table 4.3 showing the M_{ct} matrix of our running example. We focus on calculating the DICE similarities for the candidate date *2010* and for the relevant word *Port-au-Prince*. Based on the above representation and on a threshold $T > 0$ we determine the eligible context vectors for both *2010* and *Port-au-Prince*. The result will be a vector whose components are arranged in the descending order of the similarity value. As such, we obtain (*Haiti earthquake*, *Haiti*, *major earthquakes*, *catastrophic damage*, *Port-au-Prince*, *1500*, *2011*, *1564*) to *2010* and (*catastrophic damage*, *2010*, *Haiti earthquake*, *Haiti*) to *Port-au-Prince*. After defining N we may then determine the final version of the context vectors. For example, if N is

[25] i.e. that co-occur at least once with d_j.

set to 2 we will have (*Haiti earthquake*, *Haiti*) as the context vector of *2010* and (*catastrophic damage*, 2010) as the final context vector of *Port-au-Prince*.

Table 4.3: M_{ct} matrix for our running example.

	Haiti earthquake	major earthquakes	Haiti	catastrophic damage	Port-au-Prince	Concepción de la Vega	1500	1564	2010	2011
Haiti earthquake	.	.	.	.	.	.	.	.	0.8	.
major earthquakes	.	.	.	.	.	.	.	.	0.66	.
Haiti	.	.	.	.	.	.	.	.	0.8	.
catastrophic damage	.	.	.	.	.	.	.	.	0.66	.
Port-au-Prince	0.5	0	0.5	1	1	0	0	0	0.66	0
Concepción de la Vega	.	.	.	.	.	.	.	.	0	.
1500	.	.	.	.	.	.	.	.	0.66	.
1564	.	.	.	.	.	.	.	.	0.5	.
2010	.	.	.	.	.	.	.	.	1	.
2011									0.66	

IS is now ready to compute the corresponding similarity between each of the tokens, as depicted in Figure 4.4. Specifically it will compute the level of relatedness between *catastrophic damage* and the two other context tokens of *2010* - i.e. *Haiti earthquake*, *Haiti* - and then between *2010* and all other context tokens of *2010* and so on and so forth, thus promoting semantic similarity. Note that the similarity between each pair of tokens is again determined by S. We recall that this measure was already used to determine the set of best tokens that should be part of the context vectors.

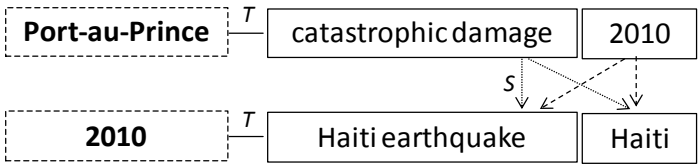


Figure 4.4: (WD;WD) context vector representation for *Port-au-Prince* and *2010*.

Next, we describe the *F* aggregation function which is used to combine the several $sim(w_j, d_j)$.

F aggregation function

In order to combine the different similarity values produced for the candidate date, d_j, in a single value capable of representing its relevance, we propose an aggregation function *F*. With that objective in mind, we consider three different *F* functions:

1. The Max/Min;
2. The Arithmetic Mean;
3. The Median.

While the Mean and the Median are measures of central tendency, the Max/Min approach relies on extreme values. In order to understand this approach more adequately, we establish two requirements: MAX and MIN.

R4.1 (MAX): the higher the number of relevant words related to the candidate date, the higher the similarity. To enter the specifics, the system selects the maximum similarity, within all the (w_j, d_j) similarity values, if the proportion of relevant words, which appear with the candidate date is above a given threshold θ. In this case, θ has experimentally been defined as *0.2*.

R4.2 (MIN): the lower the number of relevant words related to the candidate date, the lower the similarity. As such, proportion values ≤ 0.2 result in simply selecting the $sim(q, d_j)$ as a similarity value. This is often the minimum one.

The overall strategy of our query time tagging relevance model is shown in Algorithm 1.

Algorithm 1: Assign a degree of relevance to each (q, d_j) pair

Input: query q
S ← GetSnippetsFromSearchEngine(q)
For each $S_i \in S$, i = 1,..,n
    Apply Text Processing
    W_{S_i} ← Select best relevant words/multiwords in S_i
    D_{S_i} ← Select all temporal patterns in S_i
$W_S \leftarrow \cup_{i=1}^{n} W_{S_i}$
$D_S \leftarrow \cup_{i=1}^{n} D_{S_i}$
Compute M_{ct}
For each $d_j \in D_S$
    Compute $GTE(q, d_j)$
Output: $V_{GTE_{D_S}}$ relevance

The algorithm receives a query from the user, fetches related web snippets from a given search engine and applies text processing to all web snippets. This processing task involves selecting the most relevant words/multiwords and collecting the candidate years in each web snippet. Words and candidate years are then associated with a list of distinct terms. Finally, each candidate year is given a temporal similarity value computed by $GTE(q, d_j)$. The final relevance results are stored in a new vector called $V_{GTE_{D_s}}$ defined in Equation 4.10:

$$V_{GTE_{D_s}} = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_t \end{bmatrix}, \tag{4.10}$$

where $T_k, k = 1, \ldots, t$ represents the temporal similarity between a candidate date d_j, and the query q, for the *t* distinct candidate dates.

We end this section by defining some of the requirements that the GTE should fulfill:

R4.3: The more similar q and d_j are, the higher their value, with $GTE(q, d_j)$ being close to 1 if d_j frequently co-occurs with W^*.

R4.4: d_j is more relevant for q than d'_j, if $GTE(q, d_j) > GTE(q, d'_j)$.

R4.5: $GTE(q, d_j) = 0$ if and only if d_j is not associated with any of the W^* words.

In the following section, we describe the final step of our approach.

4.2.4 GTE-Class: Date Filtering

Our next step is to define an appropriate classification strategy to determine whether the candidate temporal expressions are actually relevant or not. We named it GTE-Class. In order to accomplish this objective, we suggest two approaches. The first one is to use a classical threshold-based strategy. Given a (q, d_j) pair, the system automatically classifies a date based on the following expression:

1. Relevant, if $GTE(q, d_j) \geq \lambda$,
2. Non-relevant or wrong date, if $GTE(q, d_j) < \lambda$,

where λ has to be tuned to at least a local optimum.

An illustration of this is given in Equation 4.15 for $\lambda = 0.35$. A more thorough discussion of this value, along with many more experiments, can be found in Section 4.4.

The second strategy uses a Support Vector Machine (*SVM*) learning model. For this purpose, a set of different first order and second order similarity measures are defined for each (q, d_j) pair, in line with what has been suggested by Pecina & Schlesinger [73] in the context of collocation extraction. As such, each (q, d_j) pair can be seen as a learning instance associated with the set of different characteristics, thus defining a classical learning problem.

The final set of m relevant dates for the query q is D_S^{Rel}:

$$D_S^{Rel} = \{d_1^{Rel}, d_2^{Rel}, \dots, d_m^{Rel}\}, \tag{4.11}$$

where $d_1^{Rel} < d_2^{Rel} < \cdots < d_m^{Rel}$. Note that $d_1^{Rel}$and d_m^{Rel} represent the lower and the upper temporal bounds of the query q respectively. Similarly D_{S_i} is

$$D_{S_i}^{Rel} = \{d_{1,i}^{Rel}, d_{2,i}^{Rel}, \dots, d_{u,i}^{Rel}\}, \tag{4.12}$$

meaning the set of u relevant dates $d_{j,i}$ for the query q associated with the web snippet S_i. Based on this, each snippet S_i is no longer represented by a set of candidate temporal expressions, but by a set of relevant dates. We redefine S_i as follows:

$$S_i \rightarrow (W_{S_i}, D_{S_i}^{Rel}). \tag{4.13}$$

Finally, $V_{GTE_{D_S}}$ becomes $V_{GTE_{D_S}^{Rel}}$ such that:

$$V_{GTE_{D_S}^{Rel}} = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_m \end{bmatrix}, \tag{4.14}$$

where $T_k, k = 1, \dots, m$ represents the temporal similarity between the date d_j, and the query q, for the m distinct relevant dates and $m \leq t$. This is illustrated as follows:

$$V_{GTE_{D_S}} = \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_t \end{matrix}\begin{bmatrix} 0.2 \\ 0.6 \\ 0.3 \\ 0.8 \end{bmatrix} \xrightarrow{\text{GTE-Class}} V_{GTE_{D_S}^{Rel}} = \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_t \end{matrix}\begin{bmatrix} - \\ 0.6 \\ - \\ 0.8 \end{bmatrix} = \begin{matrix} d_1 \\ d_2 \end{matrix}\begin{bmatrix} 0.6 \\ 0.8 \end{bmatrix}. \tag{4.15}$$

Note that the candidate date d_1 and d_3 are both filtered out from the final list $V_{GTE_{D_S}^{Rel}}$, as they have been classified by GTE-Class as an non-relevant temporal pattern. In the following section we define the experimental setup.

4.3 Experimental Setup

Since no benchmark for (q, d_j) pairs exists, we have built two new data sets, one based on web snippets and the other one based on query logs, both of which were made public by us for research purposes. As we aim to evaluate the temporal similarity between a query and a set of candidate dates, we need to guarantee that the queries selected are non-ambiguous in concept and temporal in its purpose, so that each query is clearly associated with a set of dates. For this purpose, we selected a set of 42 real-world text clear-concept temporal queries extracted from the 27 categories of Google Insights for Search. They are shown in Table 4.4. While the number of queries is small, it is in line with other similar works having temporal purposes. For instance, Jatowt and Yeung [47] use 50 temporal queries, Kanhabua et al. [54] base their tests on 42 queries, Jones and Diaz [51] use 50 TREC queries and finally Kanhabua & Nørvåg [53] selected 24 queries from Google Zeitgeist.

Table 4.4: List of text queries.

george bush iraq war	avatar movie	tour eiffel	steve jobs	amy winehouse
slumdog millionaire	britney spears	troy davis	waka waka	haiti earthquake
football world cup	justin bieber	adele	nissan juke	marco simoncelli
walt disney company	little fockers	swine flu	dan wheldon	volcano iceland
lena meyer-landrut	kate middleton	ryan dunn	david villa	true grit
california king bed	bp oil spill	fiat 500	Haiti	susan boyle
sherlock holmes	tour de france	lady gaga	katy perry	dacia duster
fernando alonso	david beckham	Fukushima	Obama	kate nash
osama bin laden	rebecca black			

4.3.1 Dataset Description

Based on the 42 text queries, we developed two datasets for our experiments. A web content dataset (*WC_DS*) and a query log one (*QLog_DS*). Each of the two datasets is described below.

For the WC_DS dataset we queried the Bing search engine on December 2011, collecting the top best 50 relevant web results, using for this purpose the Bing Search API, parameterized

with the *en-US* market language parameter. Of the 2100 web snippets retrieved, only those annotated with at least one candidate year term were selected. The final set consists of:

- 582 relevant web snippets S_i with years;
- 656 distinct $(S_i, d_{j,i})$ pairs, where $d_{j,i} \in D_{S_i},\ j = 1,..,t$, is a t candidate date of the snippet S_i;
- 235 distinct (q, d_j), where q is the query and d_j the candidate year.

Each query has on average 14 temporally-stamped corresponding related web snippets, which corresponds to 1.2 year references inside each web snippet. The ground truth was then obtained by automatically labeling each one of the 235 distinct (q, d_j) pairs. In order to do this we followed a twofold approach:

1. Each $(S_i, d_{j,i})$ is manually assigned a relevance label on a 2-level scale: not a date or temporally non-relevant to the query within a snippet S_i (score 0) and temporal relevant to the query within a snippet S_i (score 1). The labeler was allowed to perform a search on the web, so as to produce knowledge about the topic and eliminate context factors that might influence a change in his judgment. As the task did not seem to be prone to different judgments, we did not apply a multi-annotator scheme. The final list of judgments consists of 119 $(S_i, d_{j,i})$ labeled with score 0, and 537 with score 1.
2. Each (q, d_j) pair is then automatically labeled based on Equation 4.16:

$$(q, d_j) = \begin{cases} 1, if\ \#Rel \geq \#\overline{Rel} \\ 0, if\ \#Rel < \#\overline{Rel} \end{cases}, \tag{4.16}$$

where $\#Rel$ represents the number of $d_{j,i}$ whose relevance judgments equals to 1 in S_i and $\#\overline{Rel}$ represents the number of $d_{j,i}$ whose relevance judgments are 0 in S_i. An illustrative example is shown in Table 4.5 for the query "*true grit*". For example, for the candidate date "*2010*", $\#Rel = 7$ and $\#\overline{Rel} = 1$. As such $(q, d_j) = 1$.

Table 4.5: (q, d_j) classification for the query "*true grit*".

q	d_j	S_i	$d_{j,i}$	(q, d_j)
True Grit	**1968**	0 , 6 , 15 , 47 , 48	1, 1, 1, 1, 1	**1**
	1969	4 , 6 , 9 , 27	1, 1, 1, 1	**1**
	1982	22	0	**0**
	2006	14	0	**0**
	2010	0 , 1 , 3 , 12 , 15 , 24 , 25 , 29	1, 1, 1, 0, 1, 1, 1, 1	**1**
	2011	5 , 37	0, 0	**0**

As for the QLog_DS dataset we used the Google and Yahoo! auto-completion engines, which suggest a set of ten expanded queries for any given query with a new data extraction method. So, to enable a fair comparison, for each of the 42 text queries we tried to obtain the highest number of possible dates from completions. For this, we use three different query combinations: (a) "*query"*, (b) "*query 1"* and (c) "*query 2",* which enable us to capture the query together with candidate dates starting at 1 and 2 respectively. An example of this is given in Figure 4.5 for the query "*avatar movie*".

avatar movie	avatar movie 1	avatar movie 2
avatar movie	avatar movie 1	avatar movie 2
avatar movie **online**	avatar movie **1080p download**	avatar movie **2009**
avatar movie **cast**	avatar movie **1080p**	avatar movie **2k**
avatar movie **review**	avatar movie **15 minute preview**	avatar movie **2010**
avatar movie **download**	avatar movie **10023**	avatar movie **2012**
avatar movie **summary**	avatar movie **10012**	avatar movie 2 **release date**
avatar movie **characters**	avatar movie **10003**	avatar movie 2 **trailer**
avatar movie **free online**	avatar movie 1 **billion**	avatar movie **2009 watch online**
avatar movie **quotes**	avatar movie **15 minutes**	avatar movie **2011**
avatar movie **part 1**	avatar movie **16-minute preview**	avatar movie **2009 watch online free**

Figure 4.5: Google suggestion for the query "*avatar movie*".

Like for the previous approach, candidate dates were extracted based on the rule-based model, introduced in Section 2.3. Each (q, d_j) pair was then manually labeled in the same way as for the first dataset. Statistics of both data sets are summarized in Table 4.6 for the 42 queries. The annotation "*1*" means a relevant date, while "*0*" means an incorrect or non-relevant one.

Table 4.6: Statistics of WC_DS and QLog_DS datasets.

		#Dates	#Distinct Dates	# (q, d_j) pairs	0	1
WC_DS	**Web Snippets**	702	73	235	86	149
QLog_DS	**Google Logs**	235	39	283	98	185
	Yahoo Logs	298	74	298	105	193

4.3.2 Baseline Measures

In this section, we describe the different baseline measures and introduce the notations used in our experiments. We specially focus on corpus-based similarity measures as they are language-independent and do not require external knowledge databases. In order to achieve this, we considered nine different first order association measures, divided in two groups: those based on word co-occurrences, and those based on web hit counts. The Pointwise Mutual Information (*PMI*) [25], the Dice coefficient [37], the Jaccard coefficient [45] and the Symmetric Conditional Probability (*SCP*) [81] constitute the first group. While PMI tends to favor less co-occurrences, SCP, DICE and Jaccard give more importance to more frequent co-occurrences. These measures are defined in Equations (4.17), (4.18), (4.19) and (4.20) respectively, where $P(x,y)$ corresponds to the joint probability that terms *x* and *y* co-occur in the same web snippet, and $P(x)$ and $P(y)$ respectively correspond to the marginal probabilities that terms *x* and *y* appear in any web snippet for a given query *q*:

$$PMI(x,y) = \log_2\left(\frac{P(x,y)}{P(x).P(y)}\right), \tag{4.17}$$

$$DICE(x,y) = \frac{2\times P(x,y)}{P(x)+P(y)}, \tag{4.18}$$

$$Jaccard(x,y) = \frac{P(x,y)}{P(x)+P(y)-P(x,y)}, \tag{4.19}$$

$$SCP(x,y) = \frac{P(x,y)^2}{P(x)+P(y)}. \tag{4.20}$$

The other five similarity measures rely on the web as a corpus, by computing co-occurrences based on hit counts. This includes the Normalized Google Distance (*NGD*) [26] and four other measures collected by Bollegala et al. [13]: WebJaccard, WebOverlap, WebDice and WebPMI. These are defined in Equations (4.21), (4.22), (4.23), (4.24) and (4.25) respectively. *N* is an estimation of the number of pages indexed by a given search engine, which in the case of Google is near to 10^{10}, $h(x,y)$ returns the number of hits for the query "*x y*", $h(x)$ returns the number of hits for the query "*x*" and $h(y)$ returns the number of hits for the query "*y*":

$$NGD(x,y) = \frac{\max[\log h(x),\log h(y)-\log h(x,y)]}{\log N-\min[\log h(x),\log h(y)]}, \tag{4.21}$$

$$WebJaccard(x,y) = \frac{h(x,y)}{h(x)+h(y)-h(x,y)}, \quad (4.22)$$

$$WebOverlap(x,y) = \frac{h(x,y)}{min(h(x),h(y))}, \quad (4.23)$$

$$WebDICE(x,y) = \frac{2h(x,y)}{h(x)+h(y)}, \quad (4.24)$$

$$WebPMI(x,y) = \log_2\left(\frac{N.h(x,y)}{h(x)\times h(y)}\right). \quad (4.25)$$

Notations

In this thesis, we use the InfoSimba similarity measure as the basis of GTE. In order to evaluate our approach, we compared several versions of the GTE combined with the IS and the PMI, SCP and DICE similarity measures. Our aim is to understand its different behavior as PMI has often been preferred in the web context, as highlighted by Turney [86]. The different versions of the GTE combined with IS are represented as *IS_(X;Y)_S_F*, where (*X*;*Y*) means the representation type of the context vectors, *S* the similarity measure used in IS (PMI, SCP and DICE), whose values are registered in M_{ct} and *F* is the aggregation function that combines the different similarity values between $w_j \in W^*$ and d_j. Further experiments have been performed based on the IS measure combined with PMI, SCP and DICE, but this time without the use of any aggregation function, i.e. by exclusively taking into account query q and candidate date d_j and not their correlated words $w_j \in W^*$. Overall, all of these measures are denoted *IS_(X;Y)_S*.

All other measures will be considered as state-of-the-art metrics. In particular, we will use the first order similarity measures (PMI, SCP, DICE, Jaccard) and the web-based first order similarity measures (NgoogleDistance, WebJaccard, WebOverlap, WebDICE, WebPMI) with and without the aggregation function, denoted *S* and *S_F,* respectively.

4.3.3 Evaluation Metrics

In order to evaluate all strategies, we propose classical evaluation metrics in IR based on a confusion matrix with TP being the number of years correctly identified as relevant, TN being the number of years correctly identified as non-relevant or incorrect, FP being the number of years wrongly identified as relevant and FN being the number of years wrongly identified as non-relevant. Based on this, we calculate, as in Section 3.2, Precision, Recall, F1-Measure and Balanced Accuracy, plus *Specificity* defined in Equation 4.26:

$$Specificity\ (Spec) = \frac{TN}{TN+FP}. \quad (4.26)$$

4.4 Results and Discussion

In this section, we describe the set of experiments conducted. We test our approach over a web collection and compare the results against a query log dataset. Each experiment will be described in Section 4.4.1 and 4.4.2 respectively.

4.4.1 Experiment A

In this first set of experiments we are particularly interested in studying how GTE behaves over the WC_DS content dataset and evaluate the performance of our approach on three different aspects: (1) temporal similarity measure, (2) date filtering and (3) comparison of GTE against the baseline rule-based model, which selects all of the temporal patterns found as correct dates. In order to achieve our objectives, we conduct three experiments, denoted **A1**, **A2** and **A3**.

Experiment **A1**: Temporal Similarity Measure

First, we conducted a variety of experiments to assess the performance of the three aggregation functions: Max/Min, Mean and Median, denoted *MM*, *AM* and *M*, respectively.

The GTE similarity measure can be instantiated with different association measures of first and second order. Although its computation is direct for the first order metrics (Equation 4.17 to Equation 4.25), it requires certain configurations for the InfoSimba (Equation 4.8), namely with regard to the definition of the context vectors. In this regard, we have already defined:

- The first order association measures (PMI, SCP and DICE) to use with our second-order similarity measure IS;
- The five possible context vector representations for the two context vectors: (W;W), (D;D), (W;D), (D;W) and (WD;WD).

Yet, we must define the selection criterion for choosing the set of words and/or candidate dates to be part of the context vectors. For this, two inter-related factors should be considered:

1. The size of the context vector, denoted N;
2. A threshold similarity value, T, such that, only those values from M_{ct} with similarity value > T should be considered as possible tokens for the context vector representation.

To find an optimal combination of N and T, we evaluated the combination of each of the three different aggregation functions (Max/Min, Mean and Median), each of the three measures combined with the IS (PMI, DICE and SCP) and each of the five context vector representations ((W;W), (D;D), (W;D), (D;W), (WD;WD)). In particular, we limited the parameters within the ranges of $5 \leq N \leq +\infty$ and $0 < T \leq 0.9$ and combined them as: {*T0.0N5, T0.0N10, T0.0N20, T0.0N*+∞*, T0.05N5, T0.05N10, T0.05N20, T0.05N*+∞*,..., T0.9N5, T0.9N10, T0.9N20, T0.9N*+∞}. For example, *T0.0N*+∞ means that we are selecting as context vectors of w_j and d_j, all terms registered in M_{ct} with similarity value higher than 0, i.e. that co-occur at least one time with w_j and d_j respectively.

To identify the best combination of parameters, we measure, for each query pair, the correlation agreement between the values produced by each of the measures and the human annotations. With that in mind, we use the point biserial correlation coefficient [55], which particularly suits this task. This statistical correlation measure relates a numerical variable with a variable consisting of binary or dichotomous classifications. In our case, "*1*" represents a relevant date and "*0*" represents either a false or non-relevant date. High biserial correlation values indicate high agreement with human annotations.

Our results have shown that the best combination was achieved for *T0.05N*+∞, with a correlation value of 0.80 for the Median function, specifically for *IS_(WD;WD)_DICE_M*. This combination is denoted *BGTE* (*Best GenericTemporalEvaluation*) for the remainder of this chapter. Overall, the Median and the Mean approach offer the best results when compared to the Max/Min. Despite the fact that the Mean approach is sensitive to extreme values, its performance is quite similar to the Median function, which suggests that the IS measure has a symmetric distribution. In contrast, the Max/Min approach performs worst. This was expected given the existence of an arbitrary threshold, which causes dates to be incorrectly classified as non-relevant. It is worth noting that, irrespective of the approach, the best correlation values always occur with the IS measure. This supports the hypothesis that a second-order co-occurrence metric behaves better than a first-order similarity one. A summary of the best results for the three different aggregation functions is shown in Table 4.7 for *T0.05N*+∞.

Table 4.7: Best point biserial correlation coefficient for GTE.

Aggregation Function	Measure	$T0.05N+\infty$
Max/Min	IS_(WD;WD)_SCP_MM	0.713
Mean	IS_(WD;WD)_DICE_AM	0.799
Median	IS_(WD;WD)_DICE_M	**0.800**

In the following discussion, we show the effect of increasing the threshold T. Results presented in Table 4.8 for N+∞, show that, $T0.0$, $T0.05$ and $T0.1$ perform quite well. However, they tend to become worse as T gets increased. This is not surprising since increasing T implies a sharp reduction of the number of possible candidates for each of the two context vectors, w_j and d_j, as only relevant words and candidates dates that often co-occur with w_j and d_j, will be considered.

Table 4.8: Point biserial correlation coefficient for GTE. $0 < T \leq 0.09$. N is fixed to +∞.

Aggregation Function	$T0.0$	$T0.05$	$T0.1$	$T0.2$	$T0.03$	$T0.04$	$T0.05$	$T0.06$	$T0.07$	$T0.08$	$T0.09$
Max/Min	0.703	**0.713**	0.712	0.703	0.683	0.672	0.607	0.517	0.395	0.288	0.128
Mean	0.795	**0.799**	0.793	0.710	0.719	0.646	0.497	0.375	0.266	0.198	0.148
Median	0.799	**0.800**	0.788	0.668	0.710	0.632	0.474	0.329	0.156	0.094	0.085

While this guarantees that the two context vectors have strongly related tokens, it will naturally cause IS to perform worse. This is due to the lack of vocabulary, thereby decreasing the possibility of finding two tokens that co-occur at least once within the set of all web snippets. Indeed, we may have a pair of words w_1 and w_2 which are strongly correlated with w_j and d_j respectively, and yet IS will return a value of 0, as they never co-occur between them. A representation of this is given in Figure 4.6.

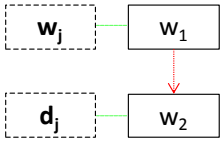


Figure 4.6: $IS(w_j,d_j)=0$.

It is also worth to note that the best biserial values often occur for *N20* and $N+\infty$ as opposed to $N5$ and $N10$. Once again, this shows that IS performs better when its context vectors contain a considerable number of tokens, as long as they guarantee a minimum value of co-

occurrence with w_j and d_j respectively. Table 4.9 shows the biserial values for $5 \leq N \leq +\infty$ when T is fixed to *0.05*.

Table 4.9: Point biserial correlation coefficient for GTE. $5 \leq N \leq +\infty$. T is fixed to *0.05*.

Aggregation Function	**N5**	**N10**	**N20**	***N*+∞**
Max/Min	0.668	0.708	0.712	**0.713**
Mean	0.550	0.724	0.795	**0.799**
Median	0.476	0.693	0.795	**0.800**

All the results are summarized in Figure 4.7, for the three different approaches, when $5 \leq N \leq +\infty$ and $0 < T \leq 0.09$.

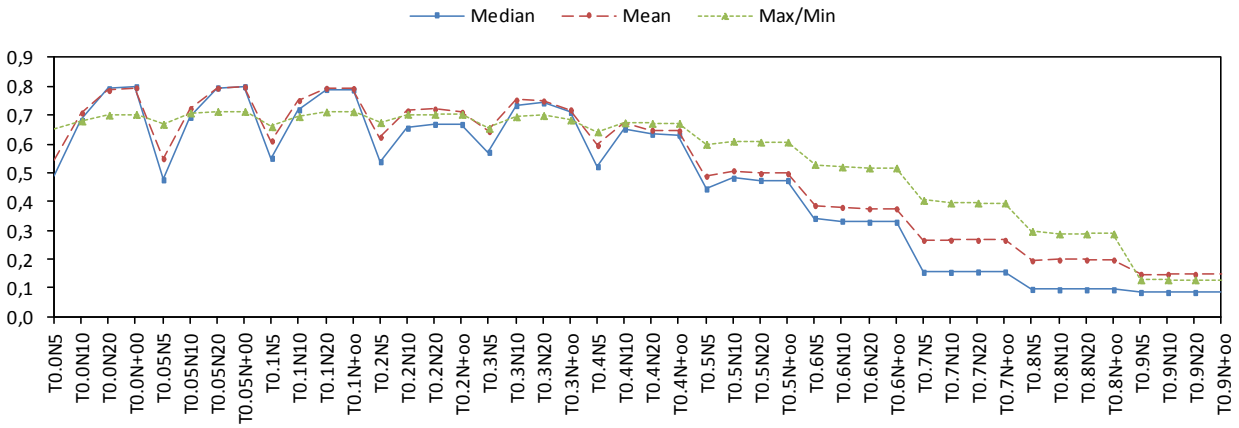


Figure 4.7: Size and threshold effect. Median, Mean and Max/Min approach. Point biserial correlation values.

A further observation led us to conclude that the type of the context vector representation greatly influences the performance of the system. We found that, regardless of the approach, the best possible representation is given by the combination of words and candidate dates, denoted (WD;WD). This is clearly depicted in Table 4.10 for *T0.05*.

Table 4.10: Best point biserial correlation coefficient for the five context vectors. *T0.05*.

Aggregation Function	**(W;W)**	**(D;D)**	**(W;D)**	**(D;W)**	**(WD;WD)**
Max/Min	0.706	0.545	0.333	0.449	**0.713**
Mean	0.768	0.358	0.387	0.149	**0.799**
Median	0.771	0.334	0.366	0.175	**0.800**

Finally, in Table 4.11, we show the similarity scores between a sub-set of (q, d_i) pairs to compare the BGTE with baseline measures. Similarity scores are normalized into a range of [0..1] for ease of comparison. The bottom row of the table shows the point biserial correlation

coefficient for each of the baseline measures. The highest correlation is reported by the proposed BGTE with a notable improvement compared to all measures, in particular to web-based ones. One reason for this situation is that web-based measures offer limited reliability when estimating term correlation due to ambiguity and the non-existence of content analysis [13]. This is a problem that tends to get even worse in a temporal context.

Table 4.11: List of classification (q, d_j) examples. BGTE vs. Baselines.

(q, d_j) Pair	Class	*BGTE*	*NGD*	*WebJaccard*	*WebDICE*	*WebPMI*	*PMI*	*DICE*	*Jaccard*	*SCP*
(True grit, 1969)	1	0.896	0.360	0.290	0.012	0.325	0.378	0.255	0.194	0.217
(True grit, 2010)	1	0.812	0.327	0.336	0.201	0.414	0.378	0.750	0.679	0.759
(Avatar movie, 2009)	1	0.670	0.325	0.516	0.621	0.455	0.261	0.412	0.330	0.214
(Avatar movie, 2011)	0	0.346	0.330	0.454	0.515	0.432	0.261	0.102	0.074	0.043
(California king bed, 2010)	1	0.893	0.334	0.398	0.388	0.417	0.518	0.329	0.257	0.287
(Slumdog millionaire, 2009)	0	0.000	0.311	0.350	0.251	0.461	0.388	0.069	0.049	0.055
(Tour Eiffel, 1512)	0	0.286	0.331	0.288	0.001	0.267	0.432	0.075	0.054	0.060
(Lady gaga, 1416)	0	0.336	0.337	0.289	0.003	0.275	0.368	0.066	0.047	0.053
(Haiti earthquake, 2010)	1	0.605	0.328	0.339	0.210	0.426	0.449	1.000	1.000	1.000
(Sherlock Holmes, 1887)	1	0.839	0.342	0.292	0.020	0.330	0.388	0.135	0.099	0.111
(Dacia duster, 1466)	0	0.096	0.323	0.288	0.000	0.206	0.378	0.067	0.048	0.054
(Waka waka, 1328)	0	0.246	0.321	0.288	0.000	0.102	0.492	0.084	0.061	0.068
(Waka waka, 2010)	1	0.944	0.328	0.332	0.188	0.420	0.492	0.742	0.670	0.749
(Bp oil spill, 2006)	0	0.277	0.300	0.350	0.248	0.454	0.545	0.094	0.068	0.076
(Bp oil spill, 2010)	1	0.838	0.328	0.323	0.154	0.426	0.254	0.384	0.304	0.211
(Volcano Iceland, 2010)	1	0.749	0.000	0.288	0.000	0.290	0.368	0.000	0.000	0.000
Point Biserial Correlation	-	0.800	-0.065	-0.110	-0.002	-0.081	-0.031	0.385	0.366	0.358

All these results support our hypothesis **H3** which states that "*Our temporal similarity measure to evaluate the degree of relation between a query and a candidate date, enables to better identify the most relevant dates related to the query*". From Table 4.11, we can also show that all the four requirements defined in Section 4.2.3 are met. For instance, requirement **R4.3** is taken into account by GTE as the similarity of the $(waka\ waka\ ,\ 2010)$ pair is close to "*1*", being that "*2010*" frequently co-occurs with all the terms in W^*, i.e., [(*fifa world cup song, 2010*); *0.922*], [(*Africa, 2010*); *0.977*], [(*shakira waka waka, 2010*); *0.961*]. Moreover, "*2009*" is more relevant to "*avatar movie*" than "*2011*", which confirms **R4.4**. Finally, the GTE similarity between the pair $(\text{slumdog millionaire},\ 2010)$ equals "*0*", meaning that no relevant word is related to "*2010*". This is easily explained by the fact that the film release was in 2008. This goes

in line with requirement **R4.5**. In the next sub-section we evaluate the performance of the date filtering schema.

Experiment **A2**: Date Filtering

The following experiment evaluates the performance of the two date filtering proposals: (1) the threshold classification and (2) the SVM classification. To accomplish this objective we define two experiments: **A2.1** and **A2.2**.

Experiment **A2.1**: Threshold-based Classification

In this first experiment, we use a classical threshold-based strategy to determine whether a date is or not relevant. In order to determine the best λ we rely on classical IR metrics. To avoid over-fitting and understand the generalization of the results, we followed, similarly to what we have done in Section 3.2, a stratified *5*-fold repeated random sub-sampling validation approach for all the proposed measures with 80% of learning instances for training and 20% for testing. Table 4.12 shows the values obtained for the BGTE measure.

Table 4.12: Stratified *5*-fold repeated random sub-sampling test dataset. BGTE results.

Test Dataset (20%)	Training Cutoff	TP	TN	FP	FN	1-Specificity	R	BA	P	F1-M	AUC
D1	0.35	23	19	2	3	0.095	0.884	0.894	0.920	0.901	0.937
D2	0.35	28	17	1	1	0.055	0.965	0.954	0.965	0.965	0.962
D3	0.35	28	16	1	2	0.058	0.933	0.937	0.965	0.949	0.945
D4	0.35	26	18	1	2	0.052	0.928	0.937	0.962	0.945	0.954
D5	0.35	31	13	3	0	0.187	1.000	0.906	0.911	0.953	0.965
Average	0.35	-	-	-	-	0.089	0.942	0.926	0.945	0.943	0.953

From Table 4.12, we can observe that the BGTE measure can achieve 94.3% F1 performance, 92.6% of Balanced Accuracy (BA), 94.5% of Precision (P) and 94.2% of Recall (R) corresponding to a threshold value of $\lambda = 0.35$. This is clearly illustrated in Figure 4.8.

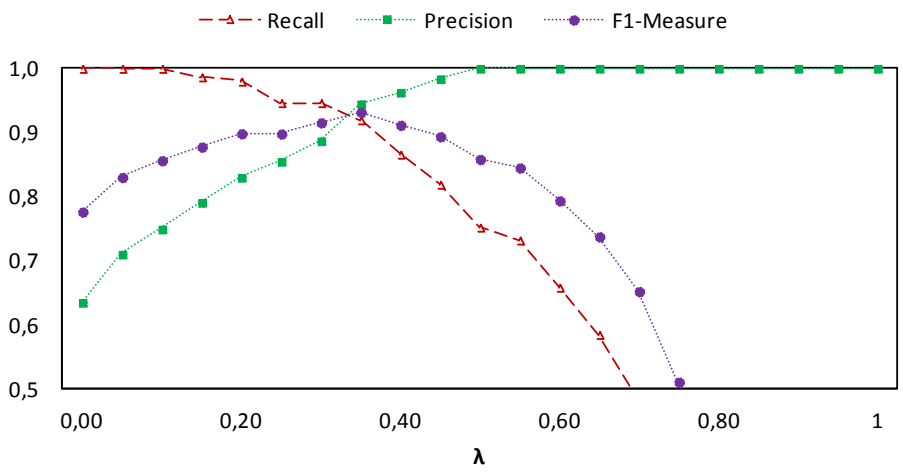


Figure 4.8: Recall, Precision and F1-M performance when varying λ for the BGTE.

These results were complemented with a Receiver Operating Characteristic (*ROC*) curve. Figure 4.9 plots this curve for the BGTE measure. The red line indicates an almost perfect classifier with an Area Under Curve (*AUC*) of 0.953 and a standard error of 0.029. The best optimization cutoff corresponds to the closest point to the upper left hand corner of the diagram, since the index of True Positives (TP) is one and of False Positives (FP) is zero. In the case of the BGTE measure, this corresponds to 0.089 of 1-Specificity and 0.942 of Sensitivity (Recall) matching a cutoff of $\lambda = 0.35$ (recall Table 4.12). Applying this λ to any retrieved results will enable to filter out non-relevant dates with high degree of accuracy. A clear example of this can be found in Table 4.11, where the candidate date "*1328*" may be considered non-relevant for the query "*waka waka*", given a GTE value of 0.246.

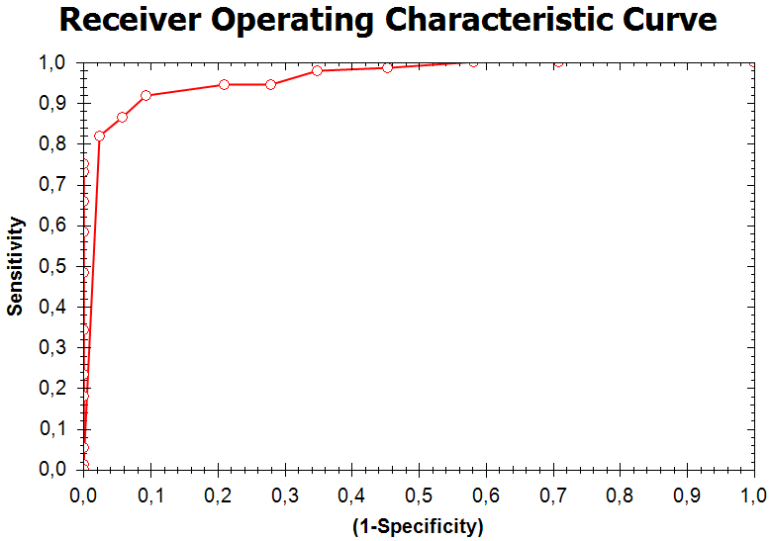


Figure 4.9: ROC curve for the BGTE measure.

A summary of the experimental results can be found in Tables 4.13, 4.14, 4.15 and 4.16, for the different measures. First, in Tables 4.13 and 4.14, we compare GTE against the baseline measures for the non-aggregated approach, and BGTE[26] against state-of-the-art metrics for the Median aggregated function, which has shown the best results.

Table 4.13: Comparative results for $sim(q, d_j)$.

Measure	Biserial	λ	1-Specificity	R	BA	P	F1-M	AUC	Error
IS_(WD;WD)_SCP	0.55	0.15	**0.064**	0.638	0.786	0.953	0.763	0.795	0.064
IS_(WD;WD)_DICE	0.56	0.15	0.107	0.754	**0.823**	0.924	**0.830**	**0.803**	**0.063**
IS_(WD;WD)_PMI	0.20	0.24	0.541	0.738	0.598	0.709	0.720	0.597	0.085
SCP	0.35	0.05	0.013	0.473	0.730	**0.986**	0.639	0.537	0.086
PMI	-0.03	0.05	0.334	0.376	0.521	0.648	0.473	0.561	0.008
DICE	0.38	0.05	0.173	0.598	0.712	0.817	0.687	0.728	0.072
Jaccard	0.36	0.05	0.119	0.526	0.703	0.885	0.659	0.696	0.007
WebPMI	-0.08	0.91	0.616	0.768	0.576	0.725	0.744	0.600	0.086
WebDice	-0.02	0.11	0.568	0.497	0.464	0.593	0.538	0.565	0.086
WebJaccard	-0.11	0.05	0.590	0.489	0.322	0.583	0.530	0.616	0.083
WebOverlap	-0.06	0.15	0.725	0.704	0.489	0.616	0.650	0.605	0.082
NGoogleDistance	0.02	0.75	0.847	**0.852**	0.502	0.580	0.690	0.529	0.085

Table 4.14: Comparative results for $F(sim(w_j, d_j))$, F= *Median*.

Measure	Biserial	λ	1-Specificity	R	BA	P	F1-M	AUC	Error
IS_(WD;WD)_SCP_M	0.67	0.25	0.239	0.932	0.846	0.896	0.898	0.891	0.046
IS_(WD;WD)_DICE_M	0.77	0.35	**0.089**	0.942	**0.926**	**0.945**	**0.943**	**0.953**	**0.029**
IS_(WD;WD)_PMI_M	0.31	0.16	0.614	0.980	0.682	0.727	0.833	0.714	0.074
SCP_M	0.10	0.05	0.661	0.890	0.614	0.652	0.748	0.578	0.085
PMI_M	0.02	0.10	0.841	**1**	0.579	0.684	0.812	0.575	0.086
DICE_M	0.30	0.15	0.619	0.958	0.669	0.723	0.823	0.656	0.079
Jaccard_M	0.35	0.10	0.422	0.881	0.729	0.792	0.833	0.769	0.067
WebPMI_M	-0.06	0.42	0.914	0.949	0.517	0.612	0.743	0.526	0.087
WebDice_M	-0.15	0.79	0.338	0.377	0.519	0.630	0.462	0.536	0.086
WebJaccard_M	0.06	0.04	0.764	0.701	0.468	0.586	0.617	0.648	0.076
WebOverlap_M	-0.11	0.90	0.635	0.630	0.483	0.640	0.619	0.551	0.008
NGoogleDistance_M	0.02	0.75	0.905	**1**	0.547	0.693	0.817	0.547	0.089

[26] i.e. IS_(WD;WD)_DICE_M

Although IS has shown an improved performance compared to other state-of-the-art measures when directly applied to a (q, d_j) pair, results were not completely satisfactory. In pursuit of the principle laid out previously, we observed that the relevance between a (q, d_j) pair is better defined if, instead of just focusing on the self-similarity, all of the information regarding existing temporal relations is increased to a higher level, namely by calculating the similarities between the several $w_j \in W^*$ and d_j. Indeed, when compared to non-aggregation and non-IS similarity measures (see Table 4.13), the BGTE can produce 19.9% F1 improvements compared to the best performing measure i.e. WebPMI with 74.4% F1-M. A general overview, for Recall, Precision and F1-M metrics can be seen in Figure 4.10.

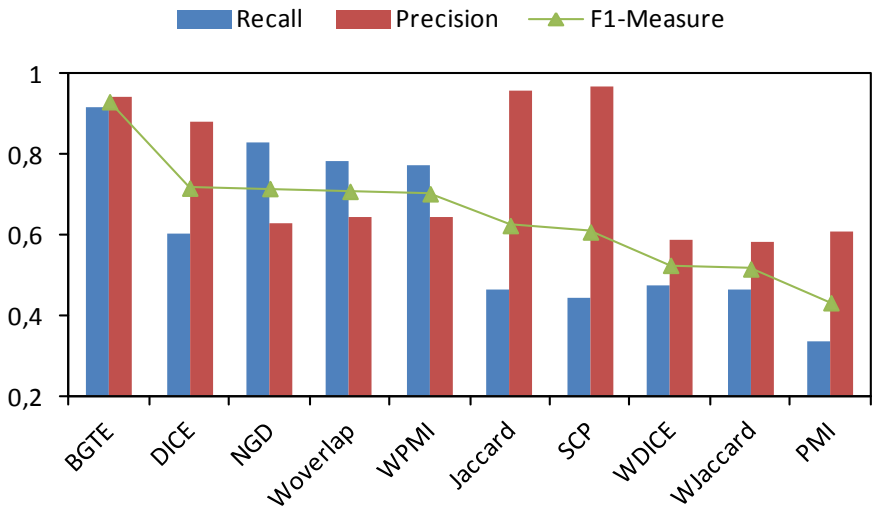


Figure 4.10: BGTE vs. Baselines.

A further observation shows that by simply adding the Median aggregator function (Table 4.14) to the simple *IS_(WD;WD)_DICE* (Table 4.13) results in an improvement of 9.8% in terms of F1-M. Indeed, all similarity measures within GTE outperform their baselines in terms of F1-M, indicating that using the Median as part of the model positively impacts the performance of the system.

Finally, we present the comparative results for the arithmetic mean and the Max/Min approach in Table 4.15 and Table 4.16. Note that on this occasion we do not present the values for the non-aggregated approach as they had already been introduced in Table 4.13 and regularly show worst results than the aggregated methodology.

Table 4.15: Comparative results for $F(sim(w_j, d_j))$, F= *Arithmetic Median*.

Measure	Biserial	1-Specificity	λ	R	BA	P	F1-M	AUC	Error
IS_(WD;WD)_SCP_AM	0.68	0.186	0.30	0.912	0.863	0.894	**0.903**	0.933	0.015
IS_(WD;WD)_DICE_AM	0.76	**0.127**	0.35	0.872	**0.926**	0.906	0.872	**0.963**	**0.011**
IS_(WD;WD)_PMI_AM	0.31	0.639	0.15	**0.993**	0.676	0.729	0.840	0.684	0.034
SCP_AM	0.22	0.755	0.05	0.979	0.612	0.691	0.811	0.606	0.037
PMI_AM	0.02	0.790	0.15	0.993	0.601	0.685	0.810	0.589	0.037
DICE_AM	0.40	0.662	0.15	0.993	0.665	0.721	0.836	0.695	0.034
Jaccard_AM	0.31	0.511	0.10	0.986	0.737	0.769	0.864	0.798	0.028
WebPMI_AM	-0.01	0.860	0.45	0.812	0.475	0.620	0.703	0.593	0.037
WebDice_AM	0.07	0.488	0.85	0.637	0.574	0.693	0.664	0.555	0.038
WebJaccard_AM	0.05	0.523	0.05	0.483	0.479	0.615	0.541	0.745	0.031
WebOverlap_AM	0.04	0.767	0.95	0.845	0.539	0.656	0.739	0.541	0.038
NGoogleDistance_AM	0.05	0.965	0.75	0.517	0.646	**1**	0.681	0.517	0.039

Table 4.16: Comparative results for $F(sim(w_j, d_j))$, F= *Max/Min*.

Measure	Biserial	1-Specificity	λ	R	BA	P	F1-M	AUC	Error
IS_(WD;WD)_SCP_MM	0.71	0.127	0.55	0.818	0.845	0.917	0.865	0.883	0.021
IS_(WD;WD)_DICE_MM	0.71	**0.081**	0.70	0.818	**0.868**	**0.945**	**0.877**	**0.895**	**0.020**
IS_(WD;WD)_PMI_MM	0.58	0.232	0.20	0.859	0.813	0.864	0.861	0.858	0.023
SCP_MM	0.50	0.244	0.05	0.859	0.807	0.859	0.859	0.835	0.025
PMI_MM	0.51	0.244	0.20	0.859	0.807	0.859	0.859	0.799	0.028
DICE_MM	0.59	0.232	0.15	0.859	0.813	0.864	0.861	0.848	0.024
Jaccard_MM	0.28	0.232	0.10	0.859	0.813	0.864	0.861	0.842	0.024
WebPMI_MM	0.38	0.790	0.60	0.953	0.581	0.676	0.791	0.523	0.038
WebDice_MM	0.10	0.267	0.75	0.691	0.711	0.817	0.749	0.732	0.032
WebJaccard_MM	0.08	0.244	0.60	0.637	0.696	0.818	0.716	0.724	0.032
WebOverlap_MM	0.22	0.558	0.95	0.647	0.702	0.633	0.640	0.649	0.035
NGoogleDistance_MM	0.36	0.860	0.05	**0.959**	0.549	0.658	0.781	0.549	0.038

From these results, we conclude that the performance of the Mean approach is quite similar to the Median. This contrasts with the Max/Min approach, which shows the worst performance. This is made clear by the difference between the *IS_(WD;WD)_DICE_MM* and the BGTE which goes up to 6.6% F1-M.

Experiment **A2.1**: SVM Classification

As an alternative to the threshold-based strategy, which uses a single similarity measure to classify any query date pair, we propose to train a SVM model [50], based on a combination of similarity measures. For this, we define a set of different first order and second order similarity measures for each (q, d_j) pair. As such, each (q, d_j) pair can be seen as a learning instance described by the different similarity measures and its manually defined class label (relevant or not relevant). Our experiments run over the implementation of the sequential minimal optimization algorithm, to train a support vector classifier, using a polynomial kernel with the default parameters in Weka[27]. A *5*-fold cross validation was performed, before and after a feature selection process, based on principal component analysis.

The learning instances were formed by the 24 similarity measures (Table 4.13 and Table 4.14) proposed in our experiments plus the manually associated class (relevant or non-relevant/incorrect date). We rely on the set of measures belonging to the Median approach as they have shown to achieve the best results. After feature selection was completed, only 14 similarity measures remained for the learning process. The results presented in Table 4.17 show a balanced accuracy of 88.6% and 90.3%, F1-M performance of 88.5% and 90.2%, and 87.6% and 89.4% of AUC respectively with and without feature selection.

Table 4.17: Best overall classification for each group of measures.

Attribute Set	Balanced Accuracy	Average *F*1-Measure	Average AUC	Relevant Date			Non-relevant Date		
				Precision	Recall	*F*1-Measure	Precision	Recall	*F*1-Measure
All Measures	0.903	0.902	0.894	0.920	0.926	0.923	0.872	0.862	0.867
All Measures after Feature Selection	0.886	0.885	0.876	0.907	0.913	0.910	0.849	0.839	0.844

This experiment allows us to conclude that feature selection may not lead to improved results. It further confirms the experiments of Pecina & Schlesinger [73], who show that a combination of measures, behaving differently, can offer better results. The evidence presented here, also shows that a unique adapted similarity measure in a threshold-based classification strategy can improve results over a classical learning process. Indeed, the results obtained by the SVM classification are worse than only using BGTE alone with $\lambda = 0.35$. In the same experimental conditions, the BGTE obtains 92.6% of accuracy (improvement of 2.3%), 94.3% *F*1-M (improvement of 4.1%) and 95.3% AUC (improvement of 5.3%).

[27] http://www.cs.waikato.ac.nz/ml/weka/ [February 25th, 2013]

Experiment **A3**: Comparison of BGTE against Baseline Rule-based Model

In our final experiment, we compare the results of BGTE with the baseline rule-based model (current standard in most of the T-IR tasks), which selects all of the temporal patterns found as correct dates (i.e. Recall = 1) within a given data set. As a consequence, for a fair comparison, we forced a Recall of 1 for the BGTE. Results are presented in Table 4.18.

Table 4.18: BGTE vs. Baseline rule-based model.

	Baseline	**BGTE**
Precision	0.634	**0.748**
Recall	1	1
F1-M	0.776	**0.856**

While the BGTE threshold strategy is forced to have a recall equal to one, it still significantly outperforms the baseline model. To assess if the difference between using the BGTE or the baseline rule-based model, for the correct classification of a (q, d_j) pair is significant, we performed the McNemar's test [64], a non-parametric method particularly suitable for non-independent dichotomous variables. The test, resulted in a Chi-squared statistic value equal to 126.130 with a p-value < 2.2e-16. This indicates that the difference of the correct date classifications is significantly different. Based on this result, we also built a confidence interval for the difference of means for paired samples between the number of misclassified dates given by the rule-based method and by the BGTE. The interval obtained [1.42; 2.30] clearly shows that the rule-based model retrieves, on average, more non-relevant or incorrect dates than the BGTE measure, with a 95% confidence level (minimum of 1.42 times more errors).

Both results corroborate hypothesis **H4** which states that "*The introduction of a classification model that is able to identify top relevant dates for any given implicit query while filtering out non-relevant ones, improves the correct classification of a query and a candidate date pair when compared to the baseline approach, which considers all the candidate dates as relevant for the query*".

4.4.2 Experiment B

In this section, we compare the BGTE measure (over a web collection of web snippets - WC_DS) against a query log approach (over the QLog_DS dataset) for the same 42 text queries. Table 4.19 presents the overall performance results both for Google (Google_QLogs), Yahoo!

(Yahoo_QLogs) and BGTE. Once again, it is important to note that, for a fair evaluation, we base the comparison on a Recall equal to 1 (by lowering the value of λ from 0.35 to 0.1).

Table 4.19: BGTE vs. Google_QLogs and Yahoo_QLogs.

	Google_QLogs	**Yahoo_QLogs**	**BGTE**
Precision	0.653	0.647	**0.748**
Recall	1	1	1
F1-M	0.790	0.786	**0.856**

The results obtained on this occasion show that BGTE achieves 85.6% of F1-M performance and 74.8% of Precision, which is significantly higher than the results achieved by each of the two completion engines. As in the previous experiment, we built a confidence interval for the difference of means, for paired samples, between the number of misclassified dates given by each of the two query log approaches and the BGTE approach. The interval obtained for GoogleQLogs is given by [1.32, 3.20] and for YahooQLogs it is [1.44, 3.47]. These intervals show that both approaches retrieve on average a significant number of non-relevant or incorrect dates when compared to BGTE, with 95% of confidence (between 1.32 and 1.44 minimum times more error). Not surprisingly, results show that query logs are able to return a great number of potential query related years, when compared to web snippets. More interestingly, however, is the fact that a large number of these temporally explicit queries consist of misleading temporal relations. One reason for this may lay in the fact that users tend to execute temporal queries embodying incorrect temporal patterns as they may not know the exact date related to the query (e.g. "*avatar movie 2012*"). These results strengthen hypothesis **H4**.

4.5 Summary

In this chapter we proposed a new temporal similarity measure, the Generic Temporal Evaluation (GTE), which allows us to employ different combinations of first order and second order similarity measures in order to compute the temporal intent(s) of (q, d_j) pairs. In particular, we have shown that the combination of the second order similarity measure InfoSimba with the DICE coefficient and the Median aggregation function, denoted BGTE, leads to better results than all the other combinations based on a threshold classification strategy where $\lambda = 0.35$ has been automatically evaluated. Our results indicate that the introduction of an additional layer of knowledge may affect the effectiveness of a broad set of T-IR systems, by retrieving a high number

of precise relevant dates. Based on this, we plan to use this new classifier as the basis for further improvements in the field of Temporal Clustering and Temporal Re-Ranking. We describe these two applications in the following chapters.

Chapter 5

Temporal Clustering

With so much information available on the web, the clustering of search results appears as a valid alternative to help users in their process of seeking information. One of the advantages of this alternative interface is to offer users a quick overview of a topic, without going through an extensive list of results. In this context, web snippet clustering appears as an interesting approach to group similar results on the basis of the retrieved result set. As shown by Zamir & Etzioni [87], in the context of ephemeral clustering, web snippets are likely to provide an adequate clustering of documents, as they contain the excerpts of documents mostly related to the query terms. The resulting data is a set of flat or hierarchical clusters generated "on-the-fly", which can be instantly used for interactive browsing purposes. Over the past few years some clustering engines have been proposed which include *iBoogie*[28], *Yippy*[29], *Carrot*[30] and *TagMySearch*[31] an evolution of *SnakeT* [41]. While all these systems present a large number of topic clusters, this chapter shows that they seldom include a temporal feature as part of the cluster description. The lack of such a time-oriented analysis makes it difficult for clustering search engines to return results with a temporal perspective. Moreover, it prevents users from becoming aware of the possible temporal structure of a given topic.

[28] http://www.iboogie.com [February 25th, 2013]
[29] http://search.yippy.com [February 25th, 2013]
[30] http://search.carrot2.org/stable/search [February 25th, 2013]
[31] http://acube.di.unipi.it/tagme_demo/tagmysearch.jsp [February 25th, 2013]

In this chapter[32], we focus on disambiguating a text query with respect to its temporal purpose by temporally clustering the obtained search results. Our method has two stages. It combines the identification of relevant temporal expressions extracted from web snippets with a clustering methodology, where documents are grouped into the same cluster if they share a common year. The resulting clusters directly reflect groups of individual years that consistently show a high connectivity to the text query.

For evaluation we use classical IR metrics and compare our approach with the Carrot web snippet clustering engine. Experiments are complemented with a user survey.

The main contributions of this chapter are: (1) a soft flat overlapping temporal clustering algorithm, where documents are highly related when they share a relevant common year; (2) a set of queries and ground-truth results made available to the research community, allowing our evaluation results to be compared with future approaches; (3) the provision of public web services so that GTE-Cluster can be tested by the research community; (4) an evaluation of our approach using several performance metrics and a comparison against a well known open-source web snippet clustering engine; and (5) a user study to validate our approach.

This chapter is structured as follows. Section 5.1 discusses the relevant literature related with this. Section 5.2 introduces our temporal ephemeral clustering algorithm. Section 5.3 presents the results and offers further discussion on them. Finally, Section 5.4 summarizes this chapter and adds some final remarks, suggesting future research directions.

5.1 Related Research

Temporal clustering is a relatively new subfield of T-IR. Within it, Mori et al. [67] and Shaparenko et al. [80] were the first to consider temporal clusters by detecting and tracking events by time. In another line of work, Jatowt et al. [48] suggest a clustering approach to summarize future-related information and a model-based clustering algorithm [47] for detecting future events based on information extracted from a text corpus. The task of clustering web search results by time, which is the focus of our research, was first introduced by Alonso et al. [2, 5]. In their first study [2], the authors assume two different clustering views: *topics* and *time*. Clustering by topics is based on traditional clustering approaches, supported on features extracted

[32] This chapter is partially based on the work published at the International Conference on Knowledge Discovery and Information Retrieval – KDIR2009 (Campos et al. 2009) and the IEEE/WIC/ACM International Conference on Web Intelligence – WIC2012 (Campos et al. 2012c).

from the title and the text snippet, whereas clustering by time relies on temporal attributes extracted from the metadata of the document and from its contents. This paper was later extended [5] by introducing a clustering algorithm called *TCluster*, where each cluster is formed by a set of documents sharing a temporal expression. The organization of the clusters along a timeline $T = \{T_d, T_w, T_m, T_y\}$, allows for the exploration of documents at different levels of granularity, namely days, weeks, months and years.

Unfortunately, none of these studies measure whether the temporal expressions found are indeed relevant or query-related. The possible exception is the research conducted by Alonso et al. [5]. However, clustering is made in a perspective of the document and not of the relevance of the date for a given query. The lack of such a solution causes systems to become highly dependent on the ability of the temporal tagger to determine the timestamp of the temporal expressions found, which, given the limitations outlined in Section 2.1.4, may compromise the quality of the clusters.

In order to overcome these shortcomings, we propose a new ephemeral clustering algorithm where documents are grouped according to a common year based on query temporal disambiguation. The advantage of our approach is that instead of considering all the temporal expressions as equally relevant, as currently common in most of the T-IR tasks [2, 5], we determine which ones are more relevant to the user text query. This results in a direct impact on the quality of the retrieved clusters, as non-relevant or wrong dates are discarded. We are aware that this is a simple direct application of the GTE-Class and that our clustering solution is, from a clustering point of view, a straightforward algorithm. In spite of that, we believe this can open up the debate and create opportunities for future research improvements.

In the following section we shall introduce our clustering algorithm.

5.2 GTE-Cluster

In this chapter, we describe our temporal clustering solution. GTE-Cluster focuses on adding top relevant temporal features to post-retrieval clustering based on Principle 5.1:

P5.1: Two snippets are temporally similar if they are highly related to the same set of dates.

Clustering snippets based on this principle poses however some challenges. On the one hand, we don't want that the search through the list of results is replaced by a search within the

set of clusters. On the other hand, we want to prevent that the user is faced with a set of non-relevant temporal clusters. Based on this, we define two requirements, which we shall call R5.1 and R5.2:

R5.1: In response to any query, a reduced number of clusters should be obtained, so that the search through the list of results is not replaced by a search within a set of clusters.

R5.2: Non-relevant or incorrect dates should be filtered out to avoid the clustering of snippets on the basis of misinterpreted temporal patterns.

In order to achieve both requirements, we rely on a clustering algorithm that involves the application of GTE-Class to filter out non-relevant dates. The simplicity of our method enables us to form clusters based on a high connectivity to snippets sharing a relevant common year. Each web snippet S_i contains a set of $D_{S_i}^{Rel}$ dates, which directly reflect the web snippet temporal purpose. Since its text can contain several different relevant temporal features, we permit overlapping: each snippet S_i may belong to a number of m clusters $C = \{C_1, C_2, \dots, C_m\}$. For example, a snippet with the text "*True Grit is a **2010** American Western film written and directed by the Coen brothers. It is the second adaptation of Charles Portis' **1968** novel of the same name, which ...*" would be placed in two main temporal clusters labeled "*1968*" and "*2010*" respectively.

The final set of clusters consists of m entities, where m is the number of relevant dates in D_S^{Rel}. A single cluster C_j, for $j = 1, \dots, m$ can be seen as a container of snippets associated with the same year. Intuitively, each C_j is labeled directly by D_S^{Rel}. The set of clusters are then sorted in ascending order by date. A future approach however, should consider a more elaborated mechanism by applying an inter-cluster and an intra-cluster ranking. This will enable to reduce the user effort thus avoiding the need to go through all the clusters and snippets to find the most relevant one.

The overall algorithm is formalized in Algorithm 2.

Algorithm 2: Determine the list of clusters for the query q

Input: query q

1: S ← GetSnippetsFromSearchEngine(q)
2: Compute $GTE(q, d_j)$, j = 1,.., t, candidate years (Equation 4.7)
3: D_S^{Rel} ← Determine the final list of m relevant dates by applying GTE-Class
4: $C_j \leftarrow \emptyset$
5: For each $d_j \in D_S^{Rel}$, j = 1,..,m
6: For each $S_i \in S$, i = 1,..,n
7: if S_i has d_j then
8: $C_j += S_i$

Output: *C* clusters

In the next section, we evaluate our approach from an empirical viewpoint.

5.3 Results and Discussion

This section describes the experiments conducted at this stage. Our objectives are:

1. To evaluate the ability of our clustering algorithm in order to correctly identify relevant temporal clusters C_j and snippet members S_i for the query q;
2. To compare our clustering proposal with current web snippet clustering engines;
3. To assess our approach on a real web user environment.

We have conducted three sets of experiments labeled **A**, **B** and **C**. Experiment **A** uses the WC_DS dataset to evaluate the clustering accuracy of our proposal with respect to the introduction of our classification model. The second experiment, **B**, uses the same collection to compare our temporal clustering approach to the *Carrot* web snippet clustering engine. Finally, the last experiment, **C**, tests the performance of our approach on a real web user environment by conducting a user study over the same dataset. These experiments will be described in Section 5.3.1, Section 5.3.2 and Section 5.3.3 respectively.

5.3.1 Experiment A

For this first experiment we want to evaluate the clustering accuracy of our proposal:

1. Firstly, we evaluate the quality of the clusters with respect to the set of top relevant dates identified;
2. Secondly, we evaluate the quality of the snippets with respect to the cluster label.

For these, we use the WC_DS dataset, introduced in Section 4.3.1 and conduct two experiments, labeled A1 and A2.

Experiment **A1**: Evaluating the quality of the clusters

In this experiment, we wish to evaluate the potential agreement between the clusters formed and the identification of the top relevant dates. Given that each date identified as relevant can form a cluster, the task of evaluating its quality, is the task of evaluating the proper identification of top relevant dates, whose results have already been presented in Chapter 4 (Table 4.12).

As such, in this section we analyze a few specific examples. We start by comparing our approach with Alonso et al. [5]. Our purpose is to understand the impact of using the GTE-Class with respect to non-GTE approaches, which are currently dominant in state-of-the-art research on this. In order to achieve this, we use the set of 42 queries that are part of the WC_DS dataset and compare the clusters formed by GTE-Cluster to the ones that would result from selecting as relevant all the temporal patterns found. The complete list of clusters, for the set of 42 queries, is shown in Table 5.1. Cluster labels whose dates were classified as wrong or non-relevant are identified with a single strikethrough. Results show a notable improvement when the GTE-Class approach is adopted. This is in line with the results previously presented in Table 4.18 and supports hypothesis H5 which states that "*The combination of our classification model with a clustering methodology, allows for better identification of the most relevant time periods of the query*".

Table 5.1: List of GTE-Clusters (left hand side) vs. non-GTE clusters (right hand side).

george bush iraq war	**tour de france**	**steve jobs**
1946, 1990, 1991, 1995, 2000, 2001, 2002, 2003, ~~2004~~, 2005, 2009	*1903, 2009, 2010, 2011, 2012*	*1955, 1970, 1998, 2005, 2011*
slumdog millionaire	**britney spears**	**david villa**
2008	*1981, 2008*	*1981, 2008, 2011, 2012*

george bush iraq war	**tour de france**	**steve jobs**
1946, 1990, 1991,1995,2000, 2001, 2002, 2003, ~~2004~~, 2005, ~~2007~~, 2009	*~~1004~~, ~~1989~~, 1903, ~~2006~~, 2009, 2010, 2011, 2012*	*1955, 1970, ~~1976~~, 1998, 2005, ~~2008~~, 2011*
slumdog millionaire	**britney spears**	**david villa**
2008, ~~2009~~	*1981,~~1998~~, ~~2000~~, ~~2001~~, ~~2002~~, 2008, ~~2010~~, ~~2011~~*	*1981, ~~2007~~, 2008, 2011, 2012*

football world cup	**justin bieber**	**dan wheldon**
1930, 2006, 2010, ~~2011~~, ~~2012~~, 2014, 2018, 2022	*1994, 2011*	*1978, 2005, 2011*
walt disney company	**rebecca black**	**dacia duster**
1920, 1923	*1997, 2011*	*~~1180~~, 2009, 2010, 2011*
lena meyer-landrut	**kate middleton**	**waka waka**
1991, 2010, 2011	*1982, 2010, 2011*	*2010*
fernando Alonso	**david beckham**	**obama**
1981, 1988, 1990, 1991, 2005, 2006, 2011	*1975, 2006, 2007, 2011*	*1961, 1964, 2008, ~~2011~~, 2012*
sherlock holmes	**volcano iceland**	**katy perry**
1887, 2009, 2011	*1918, 2004, 2010*	*1984, 2008, 2009, 2010, 2012*
california king bed	**bp oil spill**	**haiti**
2010, 2011	*2010, 2011*	*1953, 1956, 2010*
osama bin laden	**little fockers**	**nissan juke**
1957, 2001, 2011	*2000, 2010*	*2011, 2012*
amy winehouse	**marco simoncelli**	**susan boyle**
1983, 2000, 2011	*1987, 2011*	*1961, 2009*
haiti earthquake	**avatar movie**	**ryan dunn**
2010	*2009*	*1977, 2002, 2003, 2006, 2010, 2011*
troy davis	**adele**	**lady gaga**
1969, 1989, 1991, 2011	*1988, 2006, 2008, 2009, 2011*	*1986, ~~2004~~, 2008*
swine flu	**fiat 500**	**kate nash**
2009, 2011, 2012	*1936, 1955, 1957, 1975, 2012*	*1987, 2006, 2007, 2008, 2009*
tour eiffel	**fukushima**	**true grit**
1889, 1959	*2001, 2011*	*1968, 1969, 2010*

football world cup	**justin bieber**	**dan wheldon**
~~1505~~, 1930, 2006, ~~2008~~, 2010, ~~2011~~, ~~2012~~, 2014, 2018, 2022	*1994, 2007, 2008, 2011, ~~2015~~*	*1978, ~~2004~~, 2005, 2011*
walt disney company	**rebecca black**	**dacia duster**
~~1901~~, 1920, 1923, ~~2001~~	*1997, ~~2004~~, 2011*	*~~1180~~, ~~1466~~, ~~2008~~, 2009, 2010, 2011, ~~2012~~*
lena meyer-landrut	**kate middleton**	**waka waka**
1991, 2010, 2011	*1982, 2007, 2010, 2011*	*~~1328~~, ~~1980~~, 2010, ~~2011~~*
fernando Alonso	**david beckham**	**obama**
~~1914~~, 1981, 1988, 1990, 1991, ~~2000~~, 2005, 2006, ~~2009~~, 2011	*1975, ~~2000~~, ~~2005~~, 2006, 2007, 2011*	*1961, 1964, ~~2007~~, 2008, ~~2009~~, ~~2010~~, ~~2011~~, 2012*
sherlock holmes	**volcano iceland**	**katy perry**
1887, ~~2005~~, ~~2007~~, 2009, 2011	*1918, 2004, 2010, 2011*	*1984, 2008, 2009, 2010, 2012*
california king bed	**bp oil spill**	**haiti**
~~1988~~, 2010, ~~2011~~	*~~2006~~, 2010, 2011*	*1492, 1953, 1956, ~~2005~~, 2010, ~~2011~~*
osama bin laden	**little fockers**	**nissan juke**
~~1345~~, 1957, 1988, ~~1996~~, 2001, ~~2005~~, 2011	*~~1337~~, 2000, 2010, ~~2011~~*	*2010, 2011, 2012*
amy winehouse	**marco simoncelli**	**susan boyle**
1983, 2000, 2006, ~~2007~~, ~~2008~~, ~~2009~~, ~~2010~~, 2011	*1987, 2002, 2011*	*1961, 2009, 2010, 2011*
haiti earthquake	**avatar movie**	**ryan dunn**
1564, 1701, 2010	*2009, ~~2011~~*	*1977, 2002, 2003, 2006, ~~2008~~, ~~2009~~, 2010, 2011*
troy davis	**adele**	**lady gaga**
1968, ~~1971~~, ~~1975~~, 1989, 1991, ~~2009~~, 2011	*1988, ~~2001~~, ~~2005~~, 2006, 2008, 2009, 2011*	*~~1416~~, 1986, ~~2004~~, 2008, ~~2009~~, ~~2010~~, ~~2011~~*
swine flu	**fiat 500**	**kate nash**
1981, 2009, 2011, 2012	*1936, 1955, 1975, 1977, 2007, ~~2009~~, ~~2011~~, 2012*	*1987, ~~2006~~, 2007, 2008, 2009, ~~2011~~*
tour eiffel	**fukushima**	**true grit**
~~1175~~, ~~1512~~, ~~1889~~, 1959, ~~1989~~, ~~2006~~, ~~2007~~	*~~1500~~, 2001, 2011*	*1968, 1969, 1982, 2010, ~~2011~~*

Moreover, we show that while there is a query "*George Bush Iraq war*" assigned to 11 clusters, the average number does not exceed the value of 3.40 clusters per query when using GTE-Cluster. Indeed, while topic clustering systems usually present an excessive number of

clusters this does not seem to be the case of our temporal clustering proposal, which is in line with requirement **R5.1**. This is mostly due to two reasons. On the one hand, there is a clear reduced number of dates occurring in snippets when compared to the occurrence of words, which is due to the temporal nature of the system itself. On the other hand, our clustering algorithm is built upon the identification of top relevant dates, hence the previous exclusion of some wrong or non-relevant years. More specifically, 78 out of 90 non-relevant candidate years were correctly filtered out by our system, which results in a negative class recall of 86,7%.

In the following step, we show some results retrieved by GTE-Cluster as we seek to understand better the strengths and weaknesses of our proposal. Figure 5.1 shows the results obtained for the query "*true grit*". The snapshot shows the potential of our approach in disambiguating implicit temporal queries. By looking at the figure, we can quickly identify three main temporal clusters, $\{1968, 1969, 2010\}$ showing similarity with the query. 1968 is the year when the novel was published. 1969 and 2010 are the years of the releases of the two films based on the novel, respectively.

Figure 5.1: Relevant GTE-Clusters retrieved for the query "*true grit*".

Of the six candidate years initially identified by our rule-based model, three of them $\{1982, 2006, 2011\}$ were filtered out by the GTE-class algorithm which is in line with requirement **R5.2**. These can be seen in Figure 5.2.

Figure 5.2: Non-relevant GTE-Clusters not retrieved for the query "*true grit*".

From Figure 5.1, we can also observe the overlapping clustering methodology, as cluster "*1968*" overlaps with "*1969*" and "*2010*" with the snippet about the Wikipedia page. While overlapping could be an interesting feature of a temporal system, it may pose, however, some problems. One particular case is when documents contain a large number of dates, for which there are no further associated snippets that would enable to form a consistent cluster. A clear example is given below, for the query "*Fernando Alonso*" and one of its respective web snippet extracted from the WC_DS dataset.

Fernando Alonso
1988 - *1990* Karting Infant Category. Asturias Champion (won all 8 races), winner Galicia's Championship, winner Asturias Championship. *1990* - *1991* Karting Cadet Category.

Example 5.1: Overlapping problem for the query "*Fernando Alonso*".

In itself this snippet, would simply give rise to four temporal clusters, each one containing a single web snippet. In such cases, it would probably be better to fit snippets into a single main cluster by putting the snippet into the cluster with higher value determined by $GTE(q, d_j)$. The figure also depicts a further interesting problem, i.e. the detection of periods, which contrast with the detection of single dates. This will be further discussed in the future research section.

Finally, we highlight the language-independent characteristic of the system, which makes it possible to return relevant snippets from different languages. A clear example of this is given below, for the query "*David Villa*" and a snippet written in Spanish.

David Villa
Natural de Tuilla (Asturias). Nacido en *1981*, jugador profesional de futbol.

Example 5.2: Language independence shown for the query "*David Villa*".

Experiment **A2**: Evaluating the quality of the snippets

In this experiment we assess the accuracy of our clustering algorithm in correctly positioning snippets with regard to the cluster label. Since each candidate date found in a snippet can potentially originate a cluster, the task of evaluating the temporal relevance of the snippets is the task of evaluating the proper identification and significance of its dates with regard to the cluster. For this purpose, we conducted two experiments. Firstly, we compare the effect of applying GTE-Class in our clustering algorithm against the human annotator classifications. Secondly, we compare the human annotators classifications against a non-GTE approach, which selects as relevant all the temporal patterns found. For this, we relied on the 656 distinct $(S_i, d_{j,i})$ pairs obtained from WC_DS, where S_i is a given snippet and $d_{j,i}$ is any candidate date in S_i. Each $(S_i, d_{j,i})$ is manually assigned a relevance label on a 2-level scale: not a date or temporally non-relevant to the query within a snippet S_i (score 0) and temporal relevant to the query within a snippet S_i (score 1). To evaluate our proposal, we calculate F1-Measure (F1-M), Precision, Recall and Balanced Accuracy as in Section 4.3.3, based on a confusion matrix with TP being the number of the retrieved snippets that are relevant to the cluster label, TN being the number of snippets that were correctly classified as non-relevant with respect to the cluster label, and thus do not appear in the final list of the results, FP being the number of the retrieved snippets wrongly identified as relevant to the cluster label and FN being the number of relevant snippets missed by the system.

An example of this classification task is given in Table 5.2 for the query "*true grit*".

Table 5.2: $(S_i, d_{j,i})$ classification for the query "*true grit*".

C_j	S_i	$d_{j,i}$	Human Annotator	GTE Class	Non GTE
1968	True Grit is a 2010 American Western film written and directed by the Coen brothers. It is the second adaptation of Charles Portis' **1968** novel of the same name, which ...	1968	1	1	1
	True Grit is a 1969 American Western film written by Marguerite Roberts and directed by Henry Hathaway. It is the first adaptation of Charles Portis' **1968** novel True ...	1968	1	1	1

1969	True Grit is a **1969** American Western film written by Marguerite Roberts and directed by Henry Hathaway. It is the first adaptation of Charles Portis' 1968 novel True ...	1969	1	1	1
	True Grit 1969 Spanish subtitles	1969	0	1	1
2010	True Grit is a **2010** American Western film written and directed by the Coen brothers. It is the second adaptation of Charles Portis' 1968 novel of the same name, which ...	2010	1	1	1
2011	eBay: true grit dvd ... Star Wars: The Complete Saga (Blu-ray Disc, **2011**, 9-Dis c Set, Boxed Set)	2011	0	0	1
	True Grit – DVD Blue Ray Disk, 7 June, 2011	2011	1	0	1
			TP	**4**	**5**
			TN	**1**	**0**
			FP	**1**	**2**
			FN	**1**	**0**

The obtained results, following a micro-average scheme, point to 95.9% F1-M performance, 94.6% Precision, 97.1% Recall and 84.9% Balanced Accuracy for the GTE-Class approach, and 90.8% of F1-M performance, 83.2% of Precision, 100% of Recall and 50.0% of Balanced Accuracy for the non-GTE approach, suggesting the appropriateness of our solution in correctly positioning the snippets with regard to the temporal cluster. Note that although our approach performs quite well, these values result from the simple application of the GTE-Class, which is particularly tuned to determine the time of the queries upon all web snippets. As such, a new similarity measure focused on processing the relevance of each date in the context of its corresponding snippet and not on the context of the query, can be further studied in line with what has been proposed by Strötgen et al. [85] (previously described in Section 4.1).

Comparative results are summarized in Table 5.3 and show (marked as bold) statistically significant improvement of our clustering approach compared with the corresponding baseline using a matched paired one-sided t-test with $p < 0.05$, thus strengthening hypothesis H5.

Table 5.3: GTE-Cluster vs. non-GTE performance based on 656 distinct (S_i, $d_{j,i}$) pairs. Boldface indicates statistically significant improvement of the GTE-Cluster method compared with the non-GTE one using matched paired one-sided t-test with p-value < 0.05.

Approach	F1-M	P	R	BA
GTE-Cluster	**0.959**	**0.946**	**0.971**	**0.849**
non-GTE	0.908	0.832	1	0.500
Improvement	0.051	0.114	-0.029	0.349

5.3.2 Experiment B

In the second set of experiments, named **B**, we compare our proposal to the open source multi-faceted Carrot search engine.

For this experiment, we follow a twofold approach:

1. We demonstrate that our clustering algorithm is able to determine a wider number of temporal clusters when compared to Carrot;
2. We assess the behavior of Carrot in correctly identifying relevant temporal clusters and snippets, so as to compare their results with the ones obtained by our temporal approach.

We are aware that we are comparing two different types of approaches with different purposes and that this evaluation is somewhat uneven. Yet, the idea is precisely to show that a specific clustering temporal approach, based on the identification of relevant temporal expressions, is likely to benefit a wide range of implicit temporal queries, in which search engines continue to fail.

In order to mitigate this difficulty, we used the *Carrot Document Clustering Workbench*[33] which enables us to test Carrot upon the same dataset, i.e., the set of queries and texts that are part of the WC_DS dataset. To obtain Carrot results, we run each of the 42 text queries on the Workbench. For this objective, we used Lingo [70], an overlapping clustering algorithm, which is also used for Carrot live demo. In particular, we defined the *cluster count base* parameter of Lingo to 100 with the purpose of obtaining the highest possible number of temporal clusters. This parameter was combined with the *allow numeric labels*, in order to allow labels to contain numbers. As we intend to assess Carrot's temporal purpose, we only rely on the set of clusters (and its corresponding snippets) labeled with a year, either a single numeric value "*2009*", or a combination between years and text, e.g. "*1955 October*" or "*Susan Magdalene Boyle Born 1 April 1961*".

The final set of results went through an evaluation process to assess the performance of Carrot in terms of forming relevant temporal clusters. In order to achieve that, results were matched against the WC_DS ground truth dataset and compared by means of common IR metrics following a micro-average scheme. As expectable, Carrot performed worse when compared to

[33] http://project.carrot2.org/download.html [February 25th, 2013]

our temporal approach. Specifically, we can note a difference of 31.4% of F1-M performance in identifying relevant temporal clusters and of 28.1% of F1-M performance in terms of evaluating the potential agreement between the snippets and the clusters formed. It is interesting to note that much of this difference is due to the small values obtained by the recall measure. This validates our statement that a specific approach that is able to deal with temporal clusters is needed.

Table 5.4 and Table 5.5 summarize both dimensions for the GTE-Cluster and Carrot methodologies. Boldface indicates statistically significant improvement using matched paired one-sided t-test with p-value < 0.05, suggesting that GTE-Cluster is more effective in terms of clustering and snippets performances than the corresponding Carrot methodology.

Table 5.4: Clustering evaluation of GTE-Cluster and Carrot over the WC_DS dataset.

Approach	F1-M	P	R	BA
GTE-Cluster	**0.943**	**0.945**	**0.942**	**0.926**
Carrot	0.629	0.879	0.489	0.686
Improvement	0.314	0.066	0.453	0.240

Table 5.5: Snippet evaluation of GTE-Cluster and Carrot over the WC_DS dataset.

Approach	F1-M	P	R	BA
GTE-Cluster	**0.959**	**0.946**	**0.971**	**0.849**
Carrot	0.678	0.915	0.539	0.645
Improvement	0.281	0.031	0.432	0.204

In the following part, we analyze some of the clusters retrieved by GTE-Cluster and Carrot search engine. A summary is provided in Table 5.6 and Table 5.7 with the set of results retrieved for each of the 42 text queries. The anecdotal evidence of the clusters presented in both tables illustrate how GTE-Cluster is capable of retrieving a larger number of temporal clusters. Two illustrative examples are the queries "*slumdog millionaire*" and "*waka waka*", which are related to a set of relevant temporal instances, that were only identified by GTE-Cluster, specifically {2008} and {2010} which are the years of the film and music release, respectively.

Another interesting case is the query "*avatar movie*", which was tagged by Carrot with an non-relevant date, in this case "*2011*". A further example is given for the query "*osama bin laden*" for which GTE-Cluster was able to identify an additional relevant date "*2001*" when compared to Carrot. Note that the apparent lack of years in queries such as "*tour de france*" or "*football world cup*" (see Table 5.6) does not rely on some problem of date identification, but rather on the lack of temporal features retrieved by the web search API for each of the queries.

Table 5.6: Cluster list of the GTE-Cluster.

george bush iraq war *1946*, *1990*, *1991*, *1995*, *2000*, *2001*, *2002*, *2003*, ~~*2004*~~, *2005*, *2009*	**tour de france** *1903*, *2009*, *2010*, *2011*, *2012*	**steve jobs** *1955*, *1970*, *1998*, *2005*, *2011*
slumdog millionaire *2008*	**britney spears** *1981*, *2008*	**david villa** *1981*, *2008*, *2011*, *2012*
football world cup *1930*, *2006*, *2010*, ~~*2011*~~, ~~*2012*~~, *2014*, *2018*, *2022*	**justin bieber** *1994*, *2011*	**dan wheldon** *1978*, *2005*, *2011*
walt disney company *1920*, *1923*	**rebecca black** *1997*, *2011*	**dacia duster** ~~*1180*~~, *2009*, *2010*, *2011*
lena meyer-landrut *1991*, *2010*, *2011*	**kate middleton** *1982*, *2010*, *2011*	**waka waka** *2010*
fernando Alonso *1981*, *1988*, *1990*, *1991*, *2005*, *2006*, *2011*	**david beckham** *1975*, *2006*, *2007*, *2011*	**obama** *1961*, *1964*, *2008*, ~~*2011*~~, *2012*
sherlock holmes *1887*, *2009*, *2011*	**volcano iceland** *1918*, *2004*, *2010*	**katy perry** *1984*, *2008*, *2009*, *2010*, *2012*
california king bed *2010*, *2011*	**bp oil spill** *2010*, *2011*	**haiti** *1953*, *1956*, *2010*
osama bin laden *1957*, *2001*, *2011*	**little fockers** *2000*, *2010*	**nissan juke** *2011*, *2012*
amy winehouse *1983*, *2000*, *2011*	**marco simoncelli** *1987*, *2011*	**susan boyle** *1961*, *2009*
haiti earthquake *2010*	**avatar movie** *2009*	**ryan dunn** *1977*, *2002*, *2003*, *2006*, *2010*, *2011*
troy davis *1969*, *1989*, *1991*, *2011*	**adele** *1988*, *2006*, *2008*, *2009*, *2011*	**lady gaga** *1986*, ~~*2004*~~, *2008*
swine flu *2009*, *2011*, *2012*	**fiat 500** *1936*, *1955*, *1957*, *1975*, *2012*	**kate nash** *1987*, *2006*, *2007*, *2008*, *2009*
tour eiffel *1889*, *1959*	**fukushima** *2001*, *2011*	**true grit** *1968*, *1969*, *2010*

Table 5.7: Cluster list of Carrot search engine.

george bush iraq war *1991, 2001, 2002, 2003,* ~~*2004*~~	**tour de france** *2010, 2011, 2012*	**steve jobs** *1955*, *2005*, *2011*
slumdog millionaire	**britney spears**	**david villa** *1981*, *2008*
football world cup *2010*, *2014*, *2018*	**justin bieber** *1994*, *2011*	**dan wheldon** *1978*, *2005*, *2011*
walt disney company *1923*	**rebecca black** *2011*	**dacia duster** *2009*, *2011*
lena meyer-landrut *1991*, *2010*, *2011*	**kate middleton** *2011*	**waka waka**
fernando Alonso *2005*, *2006*, *2011*	**david beckham** *2006*, *2007*, *2011*	**obama** *2008*, ~~*2009*~~, ~~*2010*~~, ~~*2011*~~, *2012*
sherlock holmes *2009*, *2011*	**volcano iceland** *2010*	**katy perry** *2008*, *2010*
california king bed *2010*	**bp oil spill** *2010*	**haiti**
osama bin laden *1957*, *2011*	**little fockers** *2010*	**nissan juke** *2011*, *2012*
amy winehouse *1983*, *2000*, ~~*2008*~~, *2011*	**marco simoncelli** *1987*	**susan boyle** *1961*, *2009*
haiti earthquake *2010*	**avatar movie** *2009,* ~~*2011*~~	**ryan dunn** *1977*
troy davis *1991*, *2011*	**adele** *2011*	**lady gaga** ~~*2011*~~
swine flu *2009*	**fiat 500** *1936*, *1955*, *2007,* ~~*2011*~~, *2012*	**kate nash** *1987*, *2007*, ~~*2011*~~
tour eiffel *1889*	**fukushima** *2011*	**true grit** *1968*, *1969*, *2010,* ~~*2011*~~

5.3.3 Experiment C

To test our clustering approach in a real web user environment, we conducted a user survey. Our aim is to evaluate the ability of our temporal ephemeral clustering algorithm in correctly identifying relevant temporal clusters. Our initial idea was also to evaluate the performance of the non-GTE approach and of Carrot ephemeral clustering engine. However, as GTE-Cluster has proved to perform better than any of the two mechanisms we didn't find it necessary. For this experiment, we used the set of results comprising the WC_DS dataset (without human annotations). As such, the results shown to the users consist of the set of temporal clusters (and corresponding snippets) retrieved by our approach, together with those that were filtered out (clearly identified with a single strikethrough). An example of the information shown to the users is given in Table 5.8 for the query "*true grit*".

Table 5.8: User survey for the query "*true grit*".

q	d_j	S_i
True Grit	**1968**	Text of Snippet 0 Text of Snippet 4 Text of Snippet 5 Text of Snippet 6 Text of Snippet 12 ….
	1969	Text of Snippet 4 Text of Snippet 5 Text of Snippet 17 ….
	~~**1982**~~	Text of Snippet 22
	~~**2006**~~	Text of Snippet 14
	2010	Text of Snippet 0 Text of Snippet 1 Text of Snippet 5 ….
	~~**2011**~~	Text of Snippet 0 Text of Snippet 5 Text of Snippet 37 ….

The users were then requested to classify each query using a 5-scale score, in line with what has been suggested by Alonso et al. [5]:

- Excellent. All non-relevant snippets (and corresponding clusters) were filtered out and all the remaining ones are relevant;
- Good. The search results are very relevant but there might be better results. Most non-relevant snippets (and corresponding clusters) were filtered out and most remaining ones are relevant;
- Fair. Somewhat relevant. There are many snippets (and corresponding clusters) that are inaccurate, either remained so or were filtered out incorrectly;
- Not Relevant. The search result is not good because it contains too many wrong decisions;
- I do not know. I cannot evaluate the quality of the search results.

Each query was evaluated by 6 workers. The most frequent response was "*Excellent*" (see Figure 5.3) with an average of 4.30. Overall, the annotators obtained about 0.46 of agreement level by applying the Fleiss Kappa statistics [42]. Although this represents a low agreement between the annotators, it does not compromise the validity of the results, as disagreements mostly concern to the differentiation between classifying a query as "*Excellent*" or "*Good*" and not between "*Excellent*" or "*Fair*". This becomes evident as Kappa agreement gets improved to 0.81 if we simply divide the set of results into the class of relevant quality assessments (*Excellent* + *Good*) and the class of non-relevant quality ones (*Fair* + *Not Relevant* + *I do not know*).

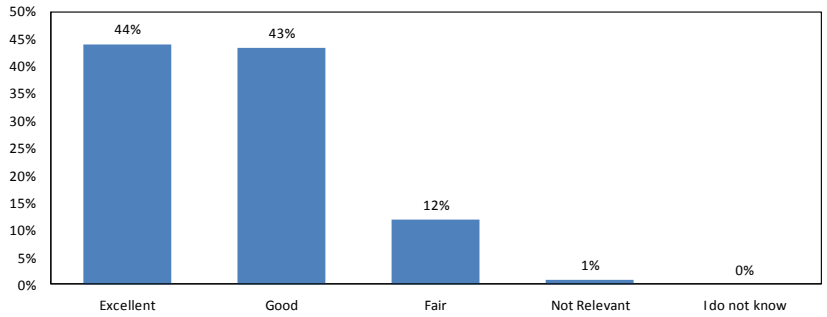


Figure 5.3: Survey results for the set of 42 queries.

An illustration of the interface of the GTE_1-Cluster web service is provided in Figure 5.4 for the query "*true grit*". The values in front of the cluster, reflect the similarity value computed by the GTE similarity measure. Note that clusters with a similarity value < 0.35 are considered non-relevant and marked in red. In contrast, relevant clusters are marked in blue. It is worth noting that our algorithm is capable of detecting as non-relevant the clusters labeled as 1870, 1960, 2011, 2012 and 2013, while detecting the most relevant ones, i.e., 1968, 1969 and 2010.

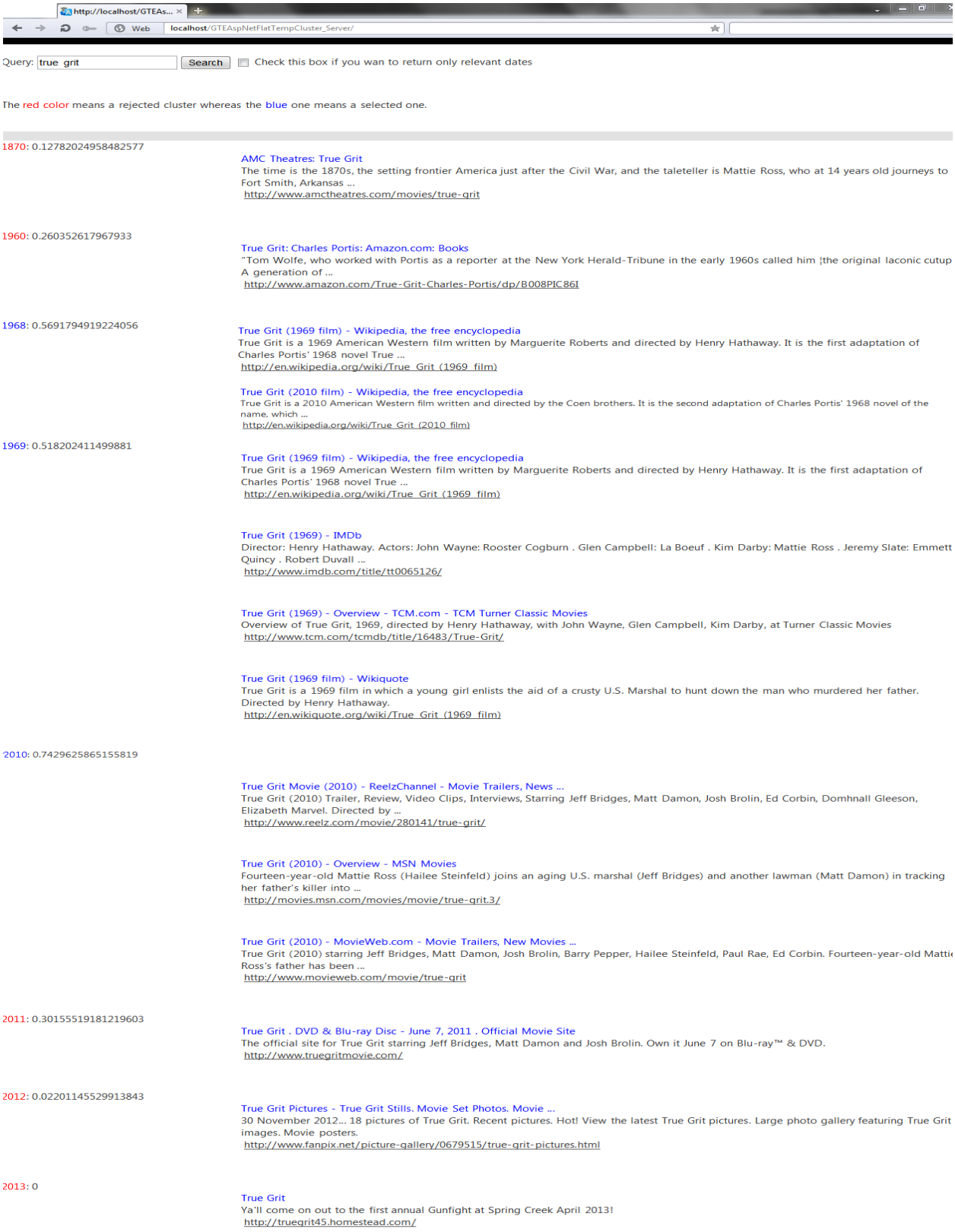


Figure 5.4: GTE-Cluster interface for the query "*true grit*". Extracted from http://wia.info.unicaen.fr/GTEAspNetFlatTempCluster_Server

5.4 Summary

In this chapter, we proposed a simple strategy for the temporal clustering of search engine query results, where snippets are clustered by year. We rely on GTE-Class, which enables us to detect top relevant years and filter out non-relevant ones. Our results show that the introduction of GTE-Class benefits the quality of the generated clusters by retrieving a high number of precise relevant dates. Comparative experiments have also been performed over Carrot ephemeral clustering engine. Results have also shown that our clustering approach is more effective than the approach of Carrot in temporally disambiguating a query, although these results were expectable. These results were complemented with a user survey showing that users mostly agree with the set of temporal clusters retrieved by our system. While we already achieved an initial stage of flat clustering by time, our proposal still lacks an approach focused on topics. This concern should be addressed in future research.

Moreover, a new similarity measure that focuses on the individual temporal processing of each snippet, in line with what has been proposed by Strötgen [85] can be further studied, so that the snippets selection process does not strictly depend on GTE-Class, which is particularly tuned to work with the set of all the snippets.

Furthermore, it is worth highlighting that the final list of snippets within each cluster simply consists of texts having at least one year annotation as we just rely upon the occurrence of explicit temporal expressions to perform clustering. While this cannot be seen as a problem, given the temporal purpose of the system, it can be improved in the future by applying a similarity measure between the words found in the snippet and each of the relevant years retrieved for the query. This is a rather simple process as similarity values are already registered in the M_{CT} conceptual temporal correlation matrix. As such, web snippets not containing any temporal expressions could be time-stamped.

Finally, an inter-cluster and intra-cluster ranking procedure should be developed to reduce the user effort when looking for relevant results. In the following chapter we focus on one of the main contributions of this thesis, the temporal re-ranking of web search results.

Chapter 6

Temporal Re-Ranking of Web Search Results

Despite the growing importance of time in Information Retrieval, most of the existing ranking functions are limited to simply returning the freshest results [11, 30, 38, 40, 57, 88]. However, freshness does not always meet the users' information needs. An example is the query "*football world cup Germany*", which confines itself to return results about the "*2006*" event, but not about the Football World Cup held in Germany in "*1974*" (see Figure 6.1).

In this chapter, we seek to re-rank the results of implicit temporal queries so as to enhance the overall temporal part of the web search results. Our ranking function *GTE-Rank* proceeds in two steps. First, we determine the time of the queries using GTE-Class. Second, we use this information to improve the retrieval effectiveness. For this purpose, we use a linear combination approach that considers topical and temporal scores, where documents are ranked to reflect the relevance of the snippet for the query, both in the conceptual and in the temporal dimension. Experiments with a publicly available dataset consisting of 1900 web snippets show that the results improve when GTE-Rank is applied. This can be very useful for a large set of underspecified queries, which although not explicitly temporally tagged, still have an inherent implicit temporal nature.

Figure 6.1: Top-10 results retrieved from Google for the query "*football world cup Germany*".

The contributions of this research can be summarized as follows: (1) we introduce a novel temporal re-ranking function supported on the identification of top relevant dates for queries where no temporal criteria is provided; (2) we adopt a language-independent methodology that can be applied to real-world search scenarios; (3) by using a content-based approach, we managed to return documents about a given period, as opposed to the retrieval of documents

written or published in a given date; (4) we provide public access to a set of queries, web snippets and ground-truth results which means that our evaluation outcomes can be compared with future approaches and (5) we also divulge a few web services so that GTE-Rank can be tested by the research community.

The structure of this chapter is as follows. Section 6.1 opens with a discussion of relevant literature. Section 6.2 describes our ranking function. Section 6.3 introduces experimental setup. Section 6.4 discusses our results. Finally, Section 6.5 summarizes this chapter with some final remarks and the suggestion of future research avenues.

6.1 Related Research

Most pioneering approaches to temporal ranking have attempted to improve the exploration of search results by biased ranking functions, usually by favoring more recent documents matching the user's query. One of the first works attempting to solve this problem was developed by Li & Croft [57]. In it, the authors incorporate time into both query-likelihood and relevance-based language models. Documents with a more recent creation date are assigned a higher probability. A similar research strategy was suggested by Efron & Golovchinsky [40]. In this case, the authors take into account not only the document publication time, but also the relationship between the publication time and the query. Queries with a more recent nature are thus allocated a more aggressive temporal impact factor. Similarly, Berberich et al. [11] and Zhang et al. [88] describe a re-ranking score so that fresh documents are ranked higher. The underlying assumption is that the user's intent is to find documents concerning the most recent years. Dong et al. [38] propose a retrieval system to answer breaking-news queries, where document freshness is taken into account by means of multiple temporal features, such as the timestamp or the link time. Finally, Dai et al. [30] propose a machine learning model that optimizes freshness and relevance simultaneously, where weights depend on the query's temporal profile.

The research that are most related to our approach are [10, 53, 65] given that they all integrate time into retrieval models with the aim of favoring the scores of documents matching the user's temporal intent. Specifically, Berberich et al. [10] suggest the integration of temporal expressions into a language model framework and rank documents according to the estimated probability of generating the query. Although it is an interesting approach, this model requires queries to contain an explicit temporal expression and documents to be explicitly timestamped.

Metzler et al. [65] and Kanhabua & Nørvåg [53] use, on the other hand, a time-dependent ranking algorithm that combines temporal and keyword similarities. Although considering implicit temporal queries, they lack some flexibility in determining the correct time of the query, making their adaptation difficult to some more specific contexts. Indeed, while Metzler et al. [65] requires access to a large query log which may not be always available, Kanhabua & Nørvåg [53] build upon the construction of temporal language models, which are difficult to adapt to open domain collections, as they need a training process.

In this thesis, we provide a more generic solution in terms of language independence and query coverage by following a content-based methodology that extracts temporal features from the contents of the document, in our case web snippets. We differ from previous takes on this subject in several other aspects. First, we do not make use of query logs. Second, we do not rely on the creation date of a document in order to determine the time of the queries, as it may differ significantly from its content. Third, our methodology is unsupervised as no specific training process is needed. Fourth, it is mostly language-independent as it implements a rule-based model supported by simple language-independent regular expressions to extract relevant dates from web snippets. Finally, besides estimating the degree of relevance of a temporal expression, we propose to determine whether or not a date is query relevant, thus using this information to improve the re-ranking of web search results.

6.2 GTE-Rank

In this section, we describe our temporal re-ranking algorithm. Our aim is to give higher weights to documents having relevant temporal features. Our assumption is that a document should be ranked higher if its contents are conceptually and temporally related to the query. This is formalized in the principle P6.1:

P6.1: The more a given document is correlated to the set of corresponding most relevant words and relevant dates associated with the query, the more the query will be associated with the document.

In order to give user's the chance to adjust the temporal and conceptual parts of the system, we propose a linear model where temporal and conceptual relevance values are gathered into a single ranking score. GTE-Rank is defined in Equation 6.1:

$$GTE\text{-}Rank(q, S_i) = \alpha * \sum_{j=1}^{u} GTE\left(q, d_{j,i}^{Rel}\right) + (1-\alpha) * \sum_{h=1}^{k} IS\left(q, w_{h,i}\right), \alpha \in [0,1], \quad (6.1)$$

where α is the tunning parameter setting the importance of each of the two dimensions, q is the query, $d_{j,i}^{Rel} \in D_{S_i}^{Rel}, j = 1,..,u$ is one of the u relevant dates of the snippet S_i and $w_{h,i} \in W_{S_i}, h = 1,..,k$ is one of the k most relevant words/multiwords of the snippet S_i.

Central to this ranking function is the computation of similarity. GTE gives the similarity between the query and each of the relevant dates found in the web snippet, and IS gives the similarity between the query and each of the relevant concepts found in the snippet. Note that one of the advantages of our approach relies precisely on the use of GTE. On the one hand, it enables GTE-Class to filter out the set of all non-relevant or non-date patterns from the input of the ranking module. On the other hand, it allows to dismiss non-relevant dates in the formation of the context concept vectors for the computation of IS, as both the query q and the word $w_{h,i}$ are formed by a combination of the best relevant words and best relevant dates. As a result, we expect to achieve an improvement of the effectiveness of results when compared to state-of-the-art algorithms that simply consider all temporal patterns as equally relevant dates. This will enable us, for example, to give higher relevance to a document with relevant dates as opposed to a document that only has non-relevant or incorrect date patterns. Below, we formalize the obvious requirement that the ranking function should fulfill.

R6.1: S_i is more relevant to q than S_i', if $GTE\text{-}Rank(q, S_i) > GTE\text{-}Rank(q, S_i')$.

The overall temporal ranking algorithm is formalized below. Given a text query q, the algorithm first identifies t candidate years in the set of snippets S. After this, GTE weights the association between the query and the set of t candidate years. The final list of m relevant dates results of applying GTE-Class. Each of these dates is then stored in the $V_{GTE_{D_s}^{Rel}}$ vector, together with the corresponding association weights. We then determine the M_{CT}^{Rel} matrix which gathers the DICE[34] similarities between "*word*"-"*word*", "*date*"-"*date*" and "*word*"-"*date*". M_{CT}^{Rel} follows the same structure of M_{ct} (recall Equation 4.9) except that it only considers *m* relevant dates as opposed to *t* candidate years where $m \leq t$. Each snippet S_i is then reordered according to the temporal (GTE) and conceptual (IS) biased factors. The final temporally biased ranking score is given by the sum of the cumulative values of GTE and IS weighted by $\alpha \in [0,1]$.

[34] We remind that GTE gives best results for: IS_(WD;WD)_DICE_M

Algorithm 3: Assign a degree of relevance to each (q, S_i) pair

Input: query q, alpha α

1: S ← RequestSearchEngine(q)
2: D_S ← Identify candidate years in S
3: Compute $GTE(q, d_j)$, j $= 1, .., t$, candidate years (Equation 4.7)
4: D_S^{Rel} ← Determine the final list of m relevant dates by applying GTE-Class
5: Determine $V_{GTE_{D_s}^{Rel}}$ (Equation 4.14)
6: Determine M_{CT}^{Rel}
7: For each $S_i \in S$, i = 1,..,n
8: For each $d_j \in D_{S_i}^{Rel}, j = 1, .., u$
9: $GTE \mathrel{+}= V_{GTE_{D_s}^{Rel}}(q, d_j)$
10: For each $w_h \in W_{S_i}, h = 1, .., k$
11: $IS \mathrel{+}= IS_{M_{CT}^{Rel}}(q, w_h)$ (Equation 4.8)
12: Compute $GTE\text{-}Rank(q, S_i) = \alpha * GTE + (1 - \alpha) * IS$

Output: (q, S_i) relevance for each $S_i \in$ S

In the next section, we define the experimental setup.

6.3 Experimental Setup

Since there are no available human-annotated data for temporal ranking purposes in the context of web snippets, we developed a new publicly available dataset (WCRank_DS). We rely on the same set of queries listed in Table 4.4, and selected all those queries that had at least one snippet labeled as non-relevant. This will allow us to apply some evaluation metrics that strictly depend on the existence of both relevant and non-relevant scores. The final set consists of 38 queries, which are listed in Table 6.1.

Table 6.1: List of text queries.

george bush iraq war	avatar movie	tour eiffel	steve jobs	amy winehouse
slumdog millionaire	britney spears	troy davis	waka waka	haiti earthquake
football world cup	justin bieber	adele	nissan juke	marco simoncelli
walt disney company	little fockers	volcano iceland	lena meyer-landrut	ryan dunn
david villa	true grit	bp oil spill	fiat 500	haiti
susan boyle	sherlock holmes	tour de france	lady gaga	katy perry
dacia duster	fernando alonso	david beckham	fukushima	obama
kate nash	osama bin laden	rebecca black		

6.3.1 Dataset Description

The list of 38 queries corresponds to a set of 1900 web snippets, of which 543 contain year terms (e.g. "*2006*"). Each (q, S_i) pair was then assigned a relevance label by a human judge on a 4-level scale. Our assumption is that users tend to prefer results that carry temporal features, as opposed to those that only have text as shown by Alonso et al. [6]. Based on this, a web snippet containing both temporal and conceptual information matching the query needs is considered to be extremely relevant and is labeled with a score of 3. It is worth noting that relevant snippets without year temporal information may also get a score of 3 (e.g. "*Amy Winehouse consumed a very large quantity of alcohol before dying at her London home, a pathologist said Wednesday as she declared Winehouse's demise...*" for the query "*Amy Winehouse*"). In the opposite direction, a web snippet that is not conceptually, nor temporally relevant, gets a score of 0. Similarly, web snippets having a year temporal reference may end up getting a score of 0 (e.g. "*©2011 EA Fragrances Co. Britney Spears™ is a trademark licensed to Elizabeth Arden, Inc. by Britney Brands, Inc.*" for the query "*Britney Spears*") as they are not considered to be temporally relevant.

Next, we formed two distinct datasets (see Table 6.2). The first one, designated *WCRank_DS1*, comprises only those web snippets having temporal features retrieved per each query. *WCRank_DS2*, in turn, includes the set of 50 web snippets retrieved for each query, , independently if they contain temporal features or not. Based on these two collections, we can then test the GTE-Rank performance in two different scenarios:

1. An exclusively temporal scenario;
2. A scenario involving the combination of temporal and conceptual relevance.

Table 6.2: Relevance judgments for the WCRank_DS1 and WCRank_DS2 datasets.

Relevance Grade	WCRank_DS1	WCRank_DS2
0	38	417
1	41	213
2	50	662
3	414	608
Total	**543**	**1900**

In the upcoming section we describe the baseline methods.

6.3.2 Baseline Methods

For the baseline ranking schema, we used the set of results retrieved by the Bing search engine and considered three different ranking models:

1. *BRank*: the Bing search engine initial ranking;
2. *RRank*: the Random ranking over Bing search engine results;
3. *ORank*: the Order by ascending date ranking over Bing search engine results.

This is in line with the study of Kanhabua & Nørvåg [53] who have only evaluated their approach for the Terrier search engine[35], using the BM25 probabilistic model with Generic Divergence From Randomness weighting as their retrieval model.

6.3.3 Evaluation Metrics

To measure how close the generated ranking results are to the ground truth, we used a set of well known IR metrics. In particular, we used Precision at k (*P@k*), Recall at k (*R@k*), Average Precision (*AP*), Mean Average Precision (*MAP*), R-Precision (*RP*), Reciprocal Rank (*RR*) and Discounted Cumulative Gain (*DCG@k*). All but the DCG@k are binary metrics, meaning that the ground truth needs to be re-built. Hence, for the grades in Table 6.2, scores (0, 1) are mapped to the non-relevant label, while scores (2, 3) are mapped to the relevant one.

More specifically, $P@k(q)$, measures how many relevant results are on the top-k snippets for the query q:

$$P@k(q) = \frac{\#\ of\ relevant\ snippets\ up\ to\ rank\ k\ for\ query\ q}{k}. \quad (6.2)$$

Similarly, $R@k(q)$ measures the fraction of relevant snippets for the query q that are successfully retrieved on the top-k positions:

$$R@k(q) = \frac{\#\ of\ relevant\ snippets\ up\ to\ rank\ k\ for\ query\ q}{\#\ of\ relevant\ snippets\ for\ query\ q}. \quad (6.3)$$

Another metric is Average Precision (AP), which computes the average precision for all values of k where k is the rank, n is the number of retrieved web snippets and Rel_k is a binary function evaluating the relevance of the kth ranked web snippet, equivalent to 1 if the web snippet at rank k is relevant and zero otherwise. These values can then be plotted in average precision histograms by computing for each query, the difference between the average precision of GRank

[35] http://terrier.org/ [February 25th, 2013]

and the median of the average precisions of the four ranking models (GRank, BRank, RRank and ORank). A positive precision means that the proposed ranking mechanism outperforms baseline methods. We define AP in Equation 6.4:

$$AP(q) = \frac{\sum_{k=1}^{n} P@k(q) \times Rel_k}{\#number\ of\ relevant\ snippets\ for\ query\ q}. \quad (6.4)$$

MAP is then computed to determine the effectiveness of our ranking mechanism over all the queries, where $|Q|$ is the number of queries. It is defined in Equation 6.5:

$$MAP = \frac{\sum_{q=1}^{|Q|} AP(q)}{|Q|}. \quad (6.5)$$

However, one of the problems of MAP is that it suffers from the effect of equally weighting each AP value, disregarding the number of relevant documents found in each of the queries. In order to overcome this problem, R-Precision (RP) has been introduced [28] to measure the fraction of relevant web snippets for the query q that are successfully retrieved at the Rth position in the ranking, where R is the total number of relevant documents for the query. This metric is particularly suitable in situations where there is a large number of relevant documents. We define R-Precision as in Equation 6.6:

$$R\text{-}Precision(q) = \frac{\#\ Rel_R}{R}. \quad (6.6)$$

The Mean R-Precision (*MRP*) is also computed by taking the arithmetic mean of all the R-Precision values for the set of all the queries, as defined in Equation 6.7:

$$MRP = \frac{\sum_{q=1}^{|Q|} R\text{-}Precision(q)}{|Q|}. \quad (6.7)$$

For instance taking two queries as an example, one with 10 relevant documents (6 of which retrieved in the top-10) and another one with 15 relevant documents (7 of which retrieved in the top-15), MRP would be calculated as follows:

$$MRP = \frac{\frac{6}{10} + \frac{7}{15}}{2} = 0.533. \quad (6.8)$$

Other metrics have been proposed bearing in mind the ranking position. The reciprocal rank (RR) metric is defined as the reciprocal (inverse) of the rank at which the first relevant document is retrieved [28]. Similarly to MAP and MRP, the Mean Reciprocal Rank (*MRR*) is defined as the average of the reciprocal ranks over all the queries, as defined in Equation 6.9, where $|Q|$ is the number of queries and rank_q is the rank position where the first relevant document for the query q was found.

$$MRR = \frac{1}{|Q|}\sum_{q=1}^{|Q|}\frac{1}{rank_q}. \quad (6.9)$$

Another method to summarize the effectiveness of our ranking algorithm is to use Recall-Precision graphs for all standard recall levels (from 0.0 to 1.0). This requires computing precision values for all these levels. For this purpose, we follow the interpolation method suggested by Croft et al. [28]. The precision P at any standard recall level R is defined in Equation 6.10, where S is the set of observed (R, P) points for a given query, i.e., the set of Recall/Precision values for each retrieved document.

$$P(R) = \max\{P' : R' \geq R \wedge (R', P') \in S\}. \quad (6.10)$$

In order to understand this, we provide the following example (adapted from Croft et al. [28]): we assume a document collection with 10 documents and two queries, for which there are five and three relevant documents respectively (see Table 6.3 where the grey color represents relevant ones). For the first query, we assume a retrieval system that ranks the relevant documents in the 1st, 3rd, 6th, 9th and 10th position. For the second query, we assume a retrieval system that ranks the relevant documents in the 2nd, 5th and 7th position. For each document of the two queries, we calculate $P@k$ and $R@k$. As such, we would have $P@1 = 1.0$ and $R@1 = 0.2$ for the first query and $P@1 = 0.0$ and $R@1 = 0.0$ for the second one. Similarly we would have $P@10 = 0.5$ and $R@10 = 1.0$ for the first query and $P@10 = 0.3$ and $R@10 = 1.0$ for the second one.

Table 6.3: Recall and precision values for ranking from two queries.

Recall	0.2	0.2	0.4	0.4	0.4	0.6	0.6	0.6	0.8	1.0
Precision	1.0	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5

Recall	0.0	0.33	0.33	0.33	0.67	0.67	1.0	1.0	1.0	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.33	0.43	0.38	0.33	0.3

Then, for each recall level R we select the $P@k$ where the corresponding $R@k \geq R$. This means that, if we are determining the precision P for recall level R 0.3 for the first query, we end up with a set of eight $P@k$ values regarding the last eight documents. The final value of P is then

determined by the maximum value within the selected $P@k$. A summary of all the standard recall levels for the two queries and the corresponding average is given in Table 6.4.

Table 6.4: Average recall-precision at standard recall levels using interpolation.

Recall	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
Ranking Query 1	1.0	1.0	1.0	0.67	0.67	0.5	0.5	0.5	0.5	0.5	0.5
Ranking Query 2	0.5	0.5	0.5	0.5	0.43	0.43	0.43	0.43	0.43	0.43	0.43
Average	0.75	0.75	0.75	0.59	0.47	0.47	0.47	0.47	0.47	0.47	0.47

For the first query, the interpolation precision for all standard recall levels up to 0.2 is 1.0, for recall levels 0.3 and 0.4 is 0.67 and for recall levels up to 1.0 is 0.5. For the second query, the interpolation precision for all standard recall levels up to 0.3 is 0.5 and for recall levels up to 1.0 is 0.43. The average precision values at standard recall levels for the set of all the queries is then calculated by averaging the precision values for both queries.

Finally, we use the Discounted Cumulative Gain (DCG@k) metric [46] and Normalized Discounted Cumulative Gain (*NDCG*) to measure the search result quality of the ranking function. DCG is supported on multiple levels of relevance. More to the point, it assigns high weights to documents in highly ranked positions and reduces those found in lower ranks. The formulation used here is defined in Equation 6.11 where $Rel_{i,q} \in \{0, 1, 2, 3\}$ is the relevance judgment of the *i*th ranked web snippet for query q

$$DCG@k(q) = \sum_{i=1}^{k} \frac{2^{Rel_{i,q}} - 1}{\log_2(1+i)}, \tag{6.11}$$

A higher DCG value reflects a better ranking of the results. NDCG is then normalized to a value between 0 and 1 by dividing the DCG value for the ideal ordering of DCG, as defined in Equation 6.12:

$$NDCG@k(q) = \frac{DCG@k(q)}{IDCG@k(q)}. \tag{6.12}$$

Similarly to MAP, MRP and MRR, the NDCG values are finally averaged over all the queries as in Equation 6.13:

$$NDCG@k = \frac{\sum_{q=1}^{Q} NDCG@k(q)}{Q}. \tag{6.13}$$

6.4 Results and Discussion

In this section, we describe the set of experiments conducted. Our first aim is to assess our ranking algorithm in situations that only include temporal texts, as well as in situations where both temporal and atemporal texts appear. In order to achieve this objective, we test our approach over WCRank_DS1 and WCRank_DS2 collections, respectively.

Secondly, we aim to test any possible difference that may exist when considering only the weights of relevant dates or accounting for all the candidate dates. In doing so, we test our ranking function using two different versions of the GTE, one based on $V_{GTE_{D_s}^{Rel}}$, named *GRank1*, and another one based on $V_{GTE_{D_s}}$, named *GRank2*. Each of these two versions is then compared to the three baseline methods by varying the α parameter within the ranges of $0 \leq \alpha \leq 1$.

Finally, we aim to test the GTE-Rank ability to pull up relevant documents and push down non-relevant ones. Indeed, as far as we know, up-to-now all related works mainly focused on pulling up temporally relevant web snippets not considering the impact of pulling down timely non-relevant ones. As a consequence, we define two different evaluation scenarios:

1. The first one denoted *Top*, aims to evaluate the ability of the ranking system as to gather only relevant documents on the top list of results;
2. The second one, called *Tail*, aims to evaluate the ability of the ranking system in order to push down all those non-relevant documents.

We are particularly interested in analyzing the GTE-Rank approach in the context of Tail analysis. Indeed, given that relevant documents are the dominant class, getting high scores on Top can easily be achieved by simply pushing up temporally relevant documents. Indeed, it is important to guarantee not only Top temporal effectiveness but also to ensure that non-relevant documents are pushed down. All IR metrics presented in Section 6.3.3 are thus redefined in accordance. As such, while for the Top approach, $P@k(q)$ measures how many relevant results are on the top-*k* documents, for the Tail one, it measures how many non-relevant results are on the tail-*k* documents. Similarly, $R@k(q)$, AP, MAP and MRP consider relevant documents when evaluating the Top scenario and non-relevant ones if the Tail one is being assessed. MRR, on the other hand, is redefined to $\overline{MRR}$ as in Equation 6.14, where $|Q|$ is the number of queries and $\mathrm{rank_q}$ is the rank position where the first *non-relevant* document for the query q is found:

$$\overline{MRR} = \frac{1}{|\mathrm{Q}|}\sum_{\mathrm{q}=1}^{|\mathrm{Q}|}\frac{1}{\mathrm{rank_q}}. \tag{6.14}$$

As such, contrary to all the other metrics, the analysis of the $\overline{MRR}$ results is made on the basis of the lowest values achieved. Furthermore, $NDCG@k$ and $\overline{MRR}$ are only used in the Top scenario, as using them on the Tail is meaningless. A summary of the different experiments is given in Table 6.5. In the upcoming parts, we offer a detailed account of the results obtained on our experiments.

Table 6.5: GTE-Rank experiments.

Experiment	Description	Dataset	Evaluation Scenario
A	GRank1 vs. GRank2	WCRank_DS1	Top
			Tail
B	GRank1 vs. Baseline	WCRank_DS1	Top
			Tail
C	GRank1 vs. Baseline	WCRank_DS2	Top
			Tail

6.4.1 Experiment A

In this experiment, we study the differences between applying GRank1 and GRank2 in our ranking function. In order to achieve this, we use the WCRank_DS1 dataset in two experiments, one with regard to the Top and another one related with the Tail scenario. The results show that GRank1 outperforms GRank2 for both scenarios, meaning that our ranking function performs better when the GTE-Class classification module is used. This strengthen hypothesis **H6** which states that "*A linear combination of the conceptual relevance with the determined time(s) of the query enhances the temporal nature of the web search results*". This is clearly illustrated in Figure 6.2 and Figure 6.3, where statistically significant improvement (p-value < 0.05) of the results of GRank1 over the GRank2 method, using matched paired one-sided t-test, is represented by solid markers. While higher precision scores occur in the Top evaluation scenario, the effect of GRank1 is mostly felt in the tail one. Indeed, if in the case of the Top scenario the differences between GRank1 and GRank2 are minimal, in the case of the Tail one, GRank1 gets improved results in terms of MAP and MRP performance in 0.035 and 0.061, respectively for $\alpha = 0.8$. This was somehow expected as non-relevant dates, to concentrate in the tail-*k* results, are simply filtered out by GRank1, while still considered in the case of GRank2. Note however, that the GRank2 method also performs quite well, as non-relevant dates, though not assigned a value of 0, as in the case of GRank1, are given a very low value by the GTE measure, thus contributing to mitigate a greater difference between both methods. A further observation, led us to conclude that

the temporal part of our ranking measure has a positive effect in the quality of the retrieved results since they get improved as α increases. This is particularly evident for the Tail approach, with GRank1 being improved in 0.122 and 0.129, for MAP and MRP, respectively, when α varies from 0.0 to 0.9. Interestingly, results become worse when changing the value of α to 1.0. We conclude that the best results come from the combination between the temporal factor and the conceptual one. A summary of the results is presented in Table 6.6 and 6.7. As for the remaining experiments, we simply rely on GRank1 approach (onwards denoted as *GRank* for simplicity) as it has proved to achieve the best performance results..

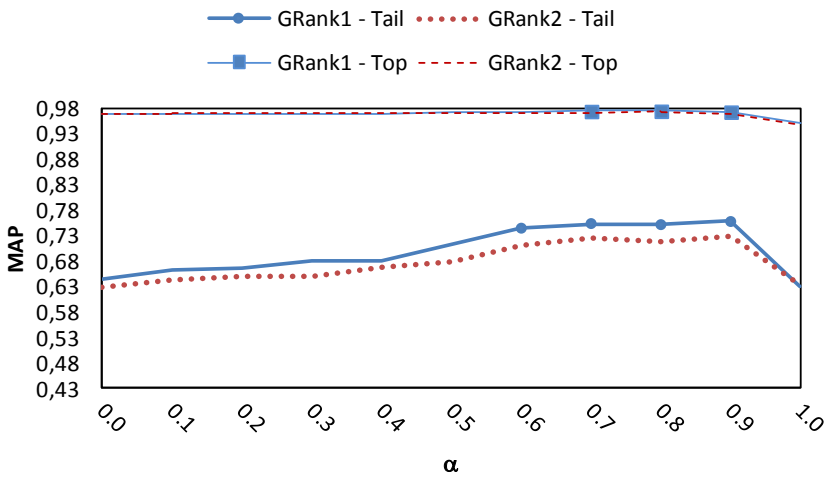


Figure 6.2: MAP. GRank1 vs. GRank2. $0.0 \leq \alpha \leq 1.0$. Top/Tail. WCRank_DS1 dataset. Solid markers indicates statistically significant improvement of the results of GRank1 over the GRank2 method using matched paired one-sided t-test with p-value < 0.05.

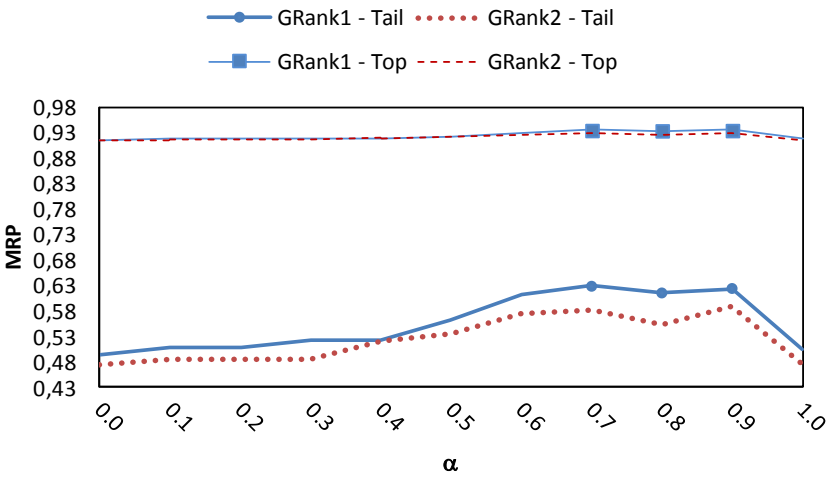


Figure 6.3: MRP. GRank1 vs. GRank2. $0.0 \leq \alpha \leq 1.0$. Top/Tail. WCRank_DS1 dataset. Solid markers indicates statistically significant improvement of the results of GRank1 over the GRank2 method using matched paired one-sided t-test with p-value < 0.05.

Table 6.6: MAP. GRank1 vs. GRank2. $0.0 \leq \alpha \leq 1.0$. Top/Tail. WCRank_DS1 dataset. Boldface indicates statistically significant improvement of the results of GRank1 over the GRank2 method using matched paired one-sided t-test with p-value < 0.05.

	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
TOP	GRank1	0.968	0.968	0.968	0.969	0.969	0.970	0.972	**0.973**	**0.974**	**0.972**	0.948
	GRank2	0.967	0.968	0.968	0.968	0.968	0.969	0.969	0.971	0.971	0.968	0.948
TAIL	GRank1	0.644	0.663	0.665	0.679	0.680	0.710	**0.743**	**0.752**	**0.750**	**0.756**	0.629
	GRank2	0.628	0.642	0.649	0.650	0.668	0.678	0.711	0.723	0.715	0.727	0.630

Table 6.7: MRP. GRank1 vs. GRank2. $0.0 \leq \alpha \leq 1.0$. Top/Tail. WCRank_DS1 dataset. Boldface indicates statistically significant improvement of the results of GRank1 over the GRank2 method using matched paired one-sided t-test with p-value < 0.05.

	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
TOP	GRank1	0.915	0.916	0.916	0.917	0.917	0.922	0.929	**0.934**	**0.933**	**0.934**	0.919
	GRank2	0.913	0.914	0.914	0.914	0.919	0.920	0.923	0.927	0.923	0.930	0.914
TAIL	GRank1	0.493	0.507	0.507	0.520	0.520	0.559	0.612	**0.627**	**0.614**	**0.623**	0.504
	GRank2	0.471	0.485	0.485	0.485	0.520	0.533	0.572	0.579	0.553	0.588	0.474

6.4.2 Experiment B

We now consider the difference between the GRank algorithm and the baseline methods when varying α from 0.0 to 1.0 over the WCRank_DS1 dataset (which consists of only temporal texts) on Top and Tail approaches. To this end, we conduct two experiments, which we designate by **B1** and **B2**. We describe their results in the two following sub-sections.

Experiment **B1**: Top

In this first experiment, we analyze the results that follow the application of GRank with regard to the Top approach. These results indicate that GRank outperforms baseline methods over the WCRank_DS1 dataset on the Top approach both for MAP (see Figure 6.4) and MRP (see Figure 6.5) metrics respectively, with statistical significance (p-value < 0.05) for α between 0.0 and 1.0 using matched paired one-sided t-test[36]. As it turns out, however, both GRank as well as the three baseline methods are able to achieve high scores, which confirms that pushing up relevant documents to the top is easy, since they constitute the dominant class. A further analysis of the

[36] Note that, to facilitate the comparison between the different methods, RRank is presented in the plot as an average of all the values. In addition, statistical significance (p-value < 0.05) of the results of GRank over each baseline method is represented by the absence of a solid marker in each of the three corresponding lines. We proceed similarly with the remaining plots.

results led us to conclude that GRank achieved the best effectiveness results, when pushing down non-relevant documents, for almost all the α degrees with statistical significance using the same test as before. This is shown in Figure 6.6. and attests to the ability of our system to ward off the non-relevant snippets from the top of the list when compared to the baseline methods. Indeed, even when compared to the baseline method with the second best performance, i.e., BRank, a difference of 0.099 (for $\alpha = 0.5$, $\alpha = 0.7$ and $\alpha = 0.8$) can still be registered in favor of GRank.

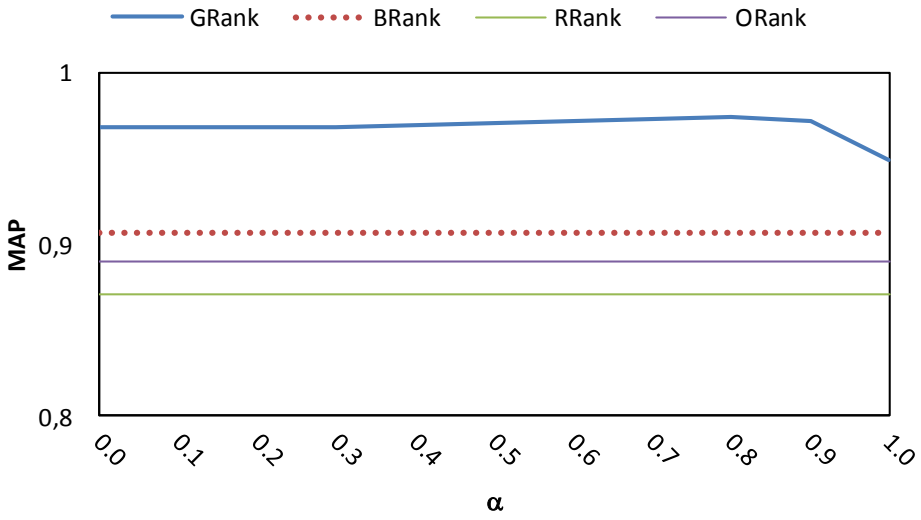


Figure 6.4: MAP. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Top. WCRank_DS1 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test

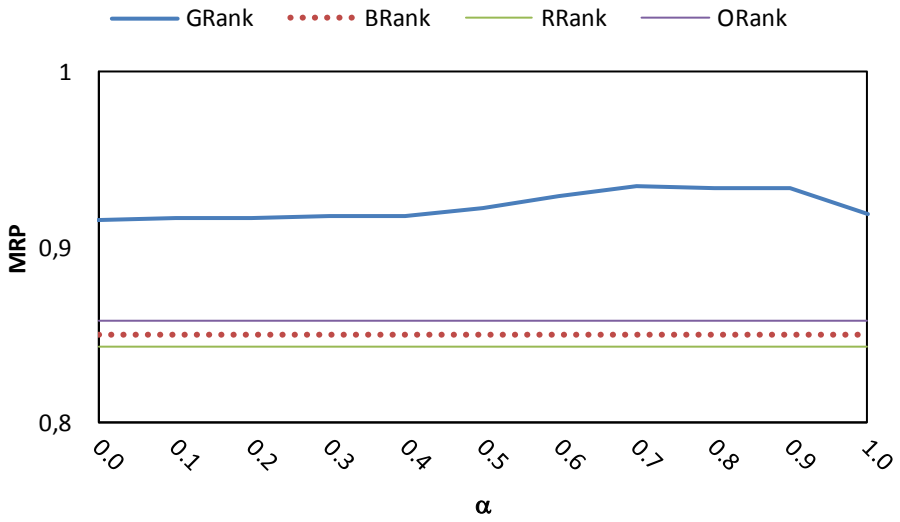


Figure 6.5: MRP. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Top. WCRank_DS1 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test

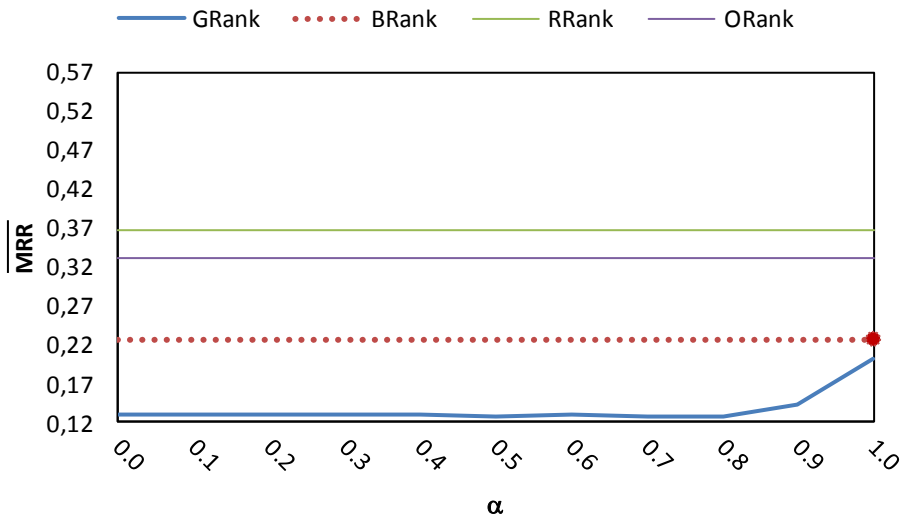


Figure 6.6: $\overline{\text{MRR}}$. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Top. WCRank_DS1 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test

A summary of the results is shown in Table 6.8 for the three different metrics, MAP, MRP and $\overline{MRR}$, plus P@1, P@3, P@5, NDCG@5, NDCG@10 and NDCG@20, which will be discussed later in more depth. Note that in almost all of the comparisons, our algorithm is statistically more significant than the corresponding baselines.

Table 6.8: MAP, MRP, $\overline{\text{MRR}}$, P@1, P@3, P@5, NDCG@5, NDCG@10 and NDCG@20 results. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Top approach. WCRank_DS1 dataset. The absence of underline indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test.

	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
MAP	GRank	0.968	0.968	0.968	0.969	0.969	0.970	0.972	0.973	0.974	0.972	0.948
	BRank	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907	0.907
	RRank	0.857	0.868	0.876	0.886	0.872	0.852	0.890	0.867	0.873	0.868	0.867
	ORank	0.889	0.889	0.889	0.889	0.889	0.889	0.889	0.889	0.889	0.889	0.889
MRP	GRank	0.915	0.916	0.916	0.917	0.917	0.922	0.929	0.934	0.933	0.934	0.919
	BRank	0.849	0.849	0.849	0.849	0.849	0.849	0.849	0.849	0.849	0.849	0.849
	RRank	0.825	0.843	0.830	0.862	0.845	0.841	0.844	0.847	0.836	0.846	0.847
	ORank	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.857
$\overline{MRR}$	GRank	0.129	0.129	0.129	0.129	0.129	0.127	0.129	0.128	0.127	0.143	0.200
	BRank	0.226	0.226	0.226	0.226	0.226	0.226	0.226	0.226	0.226	0.226	0.226
	RRank	0.406	0.367	0.336	0.330	0.352	0.435	0.285	0.403	0.338	0.383	0.394
	ORank	0.330	0.330	0.330	0.330	0.330	0.330	0.330	0.330	0.330	0.330	0.330

	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
P@1	GRank	1	1	1	1	1	1	1	1	1	0.974	0.947
	BRank	0.974	0.974	0.974	0.974	0.974	0.974	0.974	0.974	0.974	0.974	0.974
	RRank	0.842	0.737	0.895	0.763	0.895	0.737	0.921	0.947	0.921	0.921	0.895
	ORank	0.816	0.816	0.816	0.816	0.816	0.816	0.816	0.816	0.816	0.816	0.816
P@3	GRank	0.965	0.965	0.965	0.965	0.965	0.965	0.974	0.974	0.974	0.974	0.939
	BRank	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912	0.912
	RRank	0.842	0.816	0.842	0.868	0.842	0.798	0.886	0.912	0.842	0.895	0.868
	ORank	0.842	0.842	0.842	0.842	0.842	0.842	0.842	0.842	0.842	0.842	0.842
P@5	GRank	0.968	0.968	0.968	0.968	0.968	0.968	0.968	0.968	0.968	0.958	0.932
	BRank	0.889	0.889	0.889	0.889	0.889	0.889	0.889	0.889	0.889	0.889	0.889
	RRank	0.826	0.853	0.832	0.826	0.837	0.816	0.863	0.889	0.789	0.895	0.853
	ORank	0.868	0.868	0.868	0.868	0.868	0.868	0.868	0.868	0.868	0.868	0.868
NDCG@5	GRank	0.987	0.987	0.987	0.987	0.987	0.988	0.988	0.987	0.986	0.983	0.964
	BRank	0.971	0.971	0.971	0.971	0.971	0.971	0.971	0.971	0.971	0.971	0.971
	RRank	0.917	0.924	0.942	0.940	0.914	0.908	0.946	0.917	0.922	0.893	0.906
	ORank	0.920	0.920	0.920	0.920	0.920	0.920	0.920	0.920	0.920	0.920	0.920
NDCG@10	GRank	0.983	0.983	0.983	0.983	0.983	0.984	0.984	0.984	0.983	0.981	0.963
	BRank	0.957	0.957	0.957	0.957	0.957	0.957	0.957	0.957	0.957	0.957	0.957
	RRank	0.914	0.924	0.935	0.935	0.919	0.912	0.936	0.915	0.919	0.902	0.907
	ORank	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927	0.927
NDCG@20	GRank	0.981	0.981	0.981	0.981	0.981	0.982	0.982	0.982	0.981	0.980	0.964
	BRank	0.954	0.954	0.954	0.954	0.954	0.954	0.954	0.954	0.954	0.954	0.954
	RRank	0.916	0.923	0.934	0.934	0.921	0.918	0.937	0.918	0.920	0.908	0.912
	ORank	0.931	0.931	0.931	0.931	0.931	0.931	0.931	0.931	0.931	0.931	0.931

Even though GRank performs well, we still need to learn the α settings so as us to get the best performance of our system. With this end in view, we conduct nine independent cross validation rounds for the $\overline{MRR}$, MAP, MRP, P@1, P@3, P@5, NDCG@5, NDCG@10 and NDCG@20 metrics. In particular, *5*-fold cross validation operates by randomly partitioning the set of 38 queries into five folds, the first three containing 8 queries each, and the last two containing 7 queries each. Four folds are used for training, thus selecting the α that maximizes GTE-Rank and one for testing. This process is then repeated five times, using in each one, a different subset for testing and the remaining one for training. The average performance over the five folds is then used to determine the overall performance of each of the ranking models, GRank, BRank, RRank and ORank, as in Equation 6.15:

$$E = \frac{1}{n}\sum_{i=1}^{n} m(i) \ , \tag{6.15}$$

where $m(i)$ is the metric used in the cross-validation process and n is the number of folds. Results are presented in Table 6.9 for the nine metrics together with the α learned. Note that in almost all of the comparisons, our algorithm is statistically more significant than the corresponding baselines, confirming that it is possible to achieve good performance for each of the metrics by training α. Note that for the case of the $\overline{MRR}$ metric, the best value is the lowest one. A detailed analysis of the table also shows that, depending on the metric, the value of α may change significantly. From Table 6.8 we can observe that this is mostly due to the fact that the variation of the values of some metrics, irrespective of the α value, are nearly residual. This is particularly evident for the NDCG@k and for $\overline{MRR}$ metrics, for which we could have reached either a value of $\alpha = 0.6$, $\alpha = 0.7$ or $\alpha = 0.8$. As far as our GTE-Rank$_2$ web service is concerned, we rely on MAP, commonly accepted as one the most important metrics in IR, to define an α value of 0.8. However, we could have adopted an average of all the α values as well.

Table 6.9: P@k, NDCG@k, MAP, MRP and $\overline{\text{MRR}}$ results. GRank vs. Baselines. Top approach. WCRank_DS1 dataset. The absence of underline indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test.

Method	**P@1**	**P@3**	**P@5**	**NDCG@5**	**NDCG@10**	**NDCG@20**	**MAP**	**MRP**	$\overline{MRR}$
	α = 0.82	α = 0.88	α = 0.82	α = 0.58	α = 0.60	α = 0.62	α = 0.80	α = 0.78	α = 0.64
GRank	0.975	0.975	0.958	0.984	0.982	0.980	0.971	0.929	0.147
BRank	0.975	0.911	0.888	0.971	0.957	0.954	0.907	0.850	0.226
RRank	0.821	0.877	0.874	0.908	0.914	0.916	0.878	0.843	0.364
ORank	0.814	0.841	0.867	0.920	0.927	0.931	0.890	0.857	0.331

Plus, a further analysis of the results show other very interesting trends. At NDCG@5, NDCG@10 and NDCG@20 all methods show strong performances. This is not surprising, since results are heavily boosted due to a large number of relevant documents. Yet it is possible to note a difference of 0.025 for the NDCG@10, between GRank and the second best approach BRank. This is even more evident for the P@k measure, where a significant difference between GRank and the baseline methods becomes evident. Specifically, we observe a difference of 0.064 between GRank and BRank for P@3 and of 0.07 for P@5, pointing to the fact that the effect of the GRank is particularly felt after the first-*k* position. In what follows, we explore the results of P@k on a per-query basis, using average precision histograms for each query (see Figure 6.7), as explained in Section 6.3.3. Finally, we show Precision/Recall trade-off curves in Figure 6.8 based on the results presented in Table 6.10.

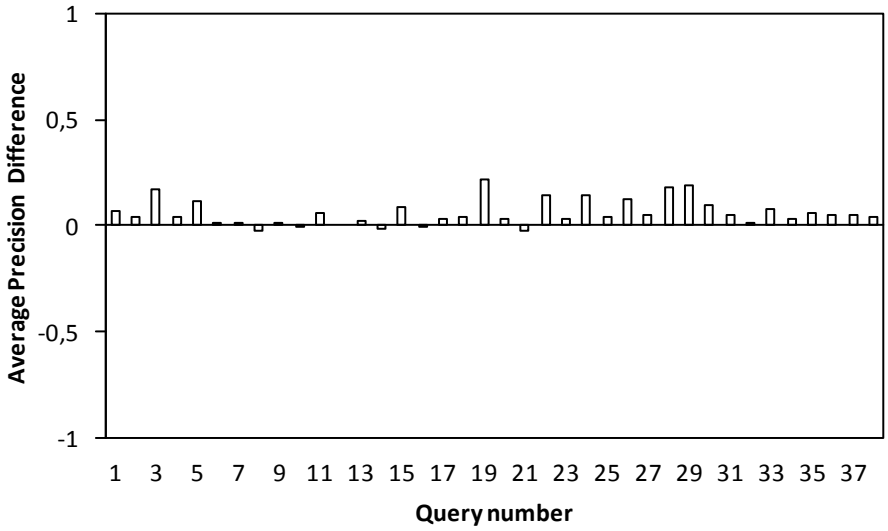


Figure 6.7: Average precision difference histogram for the 38 queries. GRank ($\alpha = 0.8$) vs. Baselines. Top. WCRank_DS1 dataset.

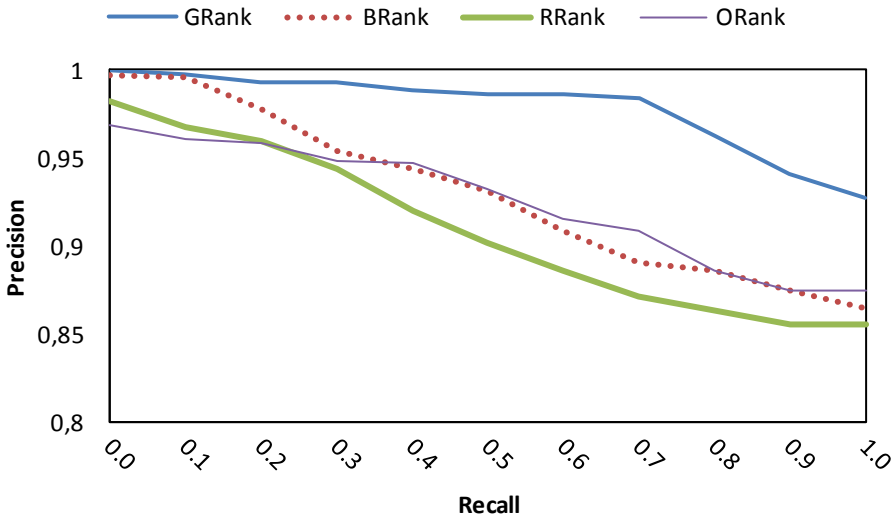


Figure 6.8: Average recall-precision for the 38 queries. GRank ($\alpha = 0.8$) vs. Baselines. Top. WCRank_DS1 dataset.

Table 6.10: Precision/Recall curve GRank ($\alpha = 0.8$) vs. Baselines. Top approach. WCRank_DS1 dataset.

Method	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
GRank	1.000	0.998	0.993	0.993	0.989	0.987	0.986	0.984	0.962	0.941	0.928
BRank	0.997	0.996	0.978	0.954	0.943	0.932	0.909	0.891	0.886	0.874	0.864
RRank	0.982	0.968	0.960	0.944	0.920	0.901	0.886	0.872	0.863	0.855	0.855
ORank	0.969	0.961	0.958	0.948	0.947	0.932	0.915	0.908	0.886	0.875	0.874

Experiment **B2**: Tail

In this section, we compare the GRank performance against the background of baseline methods over the WCRank_DS1 dataset following the Tail approach. As expected, the largest differences are mostly seen in this evaluation scenario. This is clearly depicted in Figure 6.9 and Figure 6.10 and shows the ability of GRank to push down non-relevant documents, which were originally ranked higher (BRank). More specifically, we observe a significant difference over the BRank baseline of 0.430 for the MAP metric, 0.481 for the MRP and of 0.553 for the P@1 when $\alpha = 0.9$. Table 6.11 demonstrates that, from a statistical viewpoint, GRank performs significantly better with respect to each baseline method, suggesting that our algorithm is more effective than the corresponding baseline ones.

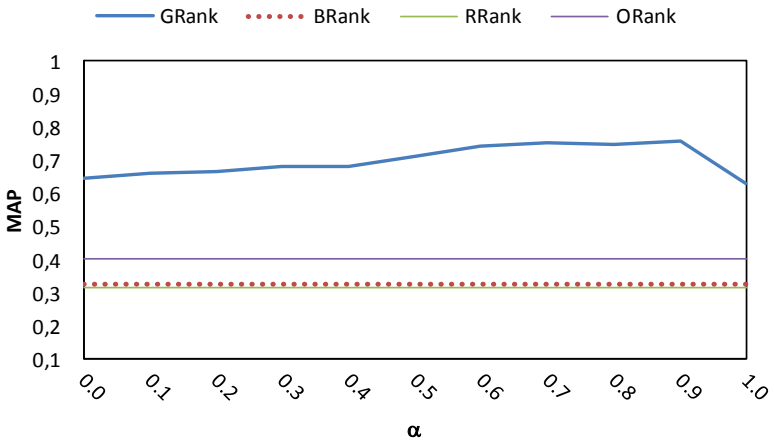


Figure 6.9: MAP. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Tail. WCRank_DS1 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test

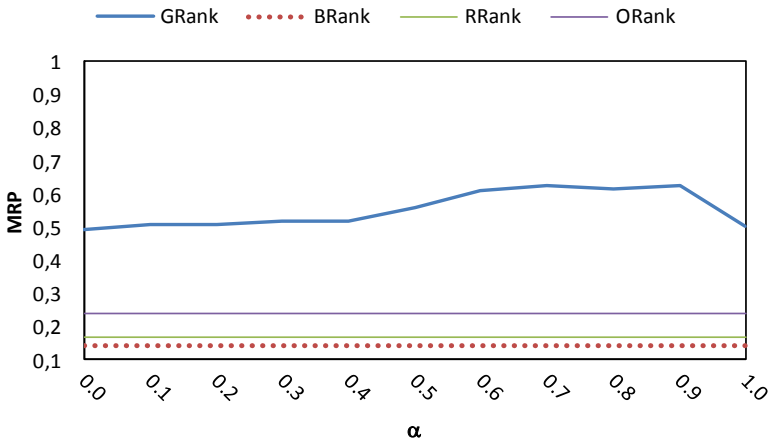


Figure 6.10: MRP. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Tail. WCRank_DS1 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test

Table 6.11: MAP, MRP, P@1, P@3 and P@5 results. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Tail approach. WCRank_DS1 dataset. All the comparisons are statistically significant with p-value < 0.05 using the matched paired one-sided t-test.

	Method	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
MAP	GRank	0.644	0.663	0.665	0.679	0.680	0.710	0.743	0.752	0.750	0.756	0.629
	BRank	0.327	0.327	0.327	0.327	0.327	0.327	0.327	0.327	0.327	0.327	0.327
	RRank	0.267	0.303	0.295	0.317	0.350	0.272	0.313	0.281	0.309	0.399	0.353
	ORank	0.405	0.405	0.405	0.405	0.405	0.405	0.405	0.405	0.405	0.405	0.405
MRP	GRank	0.493	0.507	0.507	0.520	0.520	0.559	0.612	0.627	0.614	0.623	0.504
	BRank	0.142	0.142	0.142	0.142	0.142	0.142	0.142	0.142	0.142	0.142	0.142
	RRank	0.098	0.166	0.113	0.196	0.220	0.143	0.167	0.129	0.141	0.250	0.214
	ORank	0.239	0.239	0.239	0.239	0.239	0.239	0.239	0.239	0.239	0.239	0.239
P@1	GRank	0.526	0.579	0.579	0.605	0.605	0.658	0.711	0.711	0.711	0.711	0.579
	BRank	0.158	0.158	0.158	0.158	0.158	0.158	0.158	0.158	0.158	0.158	0.158
	RRank	0.079	0.132	0.132	0.158	0.263	0.184	0.237	0.158	0.079	0.211	0.132
	ORank	0.316	0.316	0.316	0.316	0.316	0.316	0.316	0.316	0.316	0.316	0.316
P@3	GRank	0.684	0.684	0.697	0.697	0.706	0.719	0.759	0.768	0.785	0.798	0.671
	BRank	0.263	0.263	0.263	0.263	0.263	0.263	0.263	0.263	0.263	0.263	0.263
	RRank	0.329	0.184	0.228	0.206	0.303	0.246	0.351	0.272	0.184	0.263	0.228
	ORank	0.325	0.325	0.325	0.325	0.325	0.325	0.325	0.325	0.325	0.325	0.325
P@5	GRank	0.871	0.871	0.871	0.871	0.871	0.884	0.871	0.858	0.858	0.859	0.754
	BRank	0.471	0.471	0.471	0.471	0.471	0.471	0.471	0.471	0.471	0.471	0.471
	RRank	0.418	0.440	0.393	0.340	0.366	0.365	0.532	0.523	0.325	0.433	0.375
	ORank	0.521	0.521	0.521	0.521	0.521	0.521	0.521	0.521	0.521	0.521	0.521

Similarly to the Top approach, we perform *5*-fold cross validation. In particular, we conduct five independent cross validation rounds for the MAP, MRP, P@1, P@3 and P@5 metrics. Table 6.12 summarizes all these values for the Tail evaluation scenario along with the α learned. Note that GRank shows statistical significance over each baseline method, suggesting that our algorithm is more effective than the corresponding baseline ones. This is particularly evident for P@1, where GRank shows an increased performance of 0.539 compared with the BRank baseline. This clearly shows the effect of GRank in warding off the set of non-relevant documents from the tail *k* results.

Table 6.12: P@k, MAP and MRP results. GRank vs. Baselines. Tail approach. WCRank_DS1 dataset. All the comparisons are statistically significant with p-value < 0.05 using the matched paired one-sided t-test.

Method	P@1	P@3	P@5	MAP	MRP
	α = 0.90	α = 0.90	α = 0.54	α = 0.82	α = 0.74
GRank	0.696	0.798	0.869	0.737	0.593
BRank	0.157	0.260	0.472	0.324	0.139
RRank	0.186	0.211	0.313	0.275	0.116
ORank	0.311	0.327	0.527	0.404	0.239

In what follows, we provide an histogram of Average Precision difference for the of 38 queries. A summary of the results is shown in Figure 6.11, and demonstrate that GRank significantly outperforms all the baseline measures. In addition, we provide Precision/Recall curves in Figure 6.12 based on the results presented in Table 6.13. Both demonstrate that GRank is particularly suitable in pushing down non-relevant documents to the tail-*k* positions, as even for a recall of 1 it gets a precision of 0.669, 0.355 more than the BRank baseline.

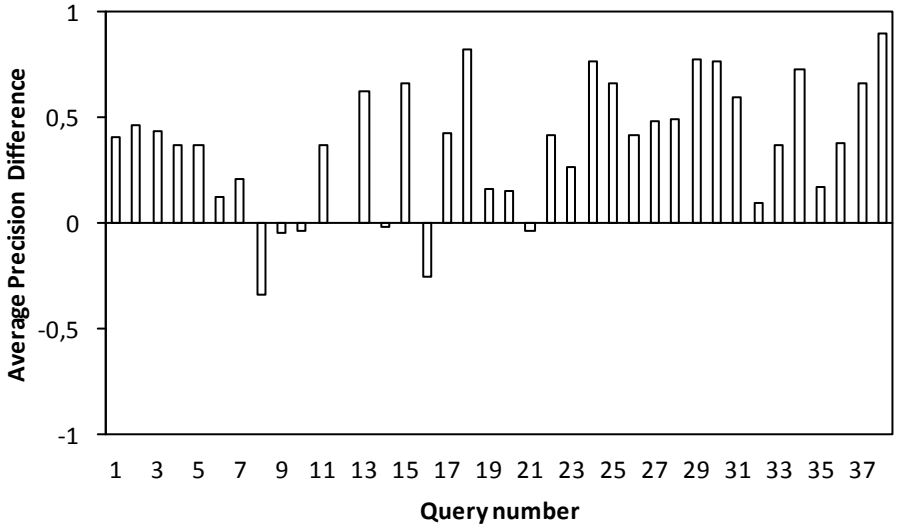


Figure 6.11: Average precision difference histogram for the 38 queries. GRank ($\alpha = 0.8$) vs. Baselines. Tail. WCRank_DS1 dataset.

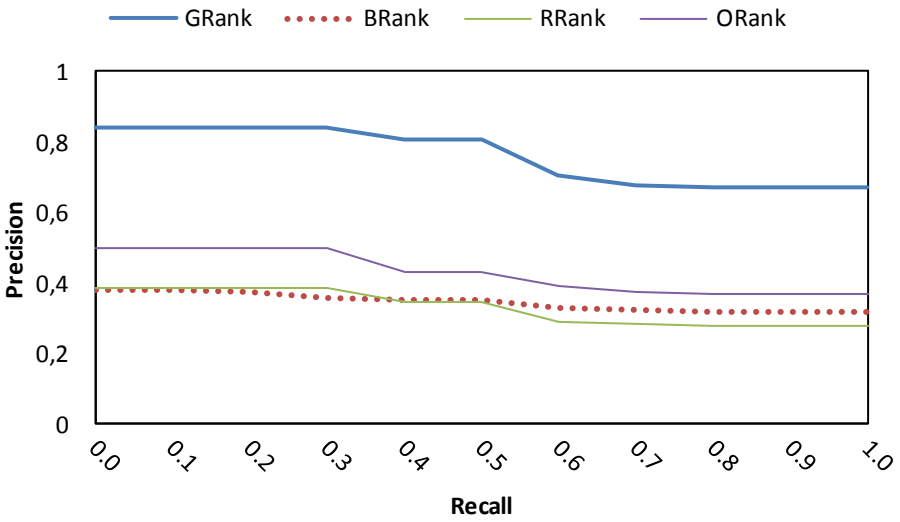


Figure 6.12: Average recall-precision for the 38 queries. GRank ($\alpha = 0.8$) vs. Baselines. Tail. WCRank_DS1 dataset.

Table 6.13: Precision/Recall. GRank ($\alpha = 0.8$) vs. Baselines. Tail. WCRank_DS1 dataset.

Method	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
GRank	0.839	0.839	0.839	0.839	0.805	0.805	0.703	0.679	0.672	0.669	0.669
BRank	0.381	0.381	0.375	0.358	0.351	0.351	0.327	0.319	0.318	0.314	0.314
RRank	0.384	0.384	0.384	0.384	0.346	0.346	0.288	0.281	0.276	0.276	0.276
ORank	0.499	0.499	0.496	0.496	0.429	0.429	0.389	0.374	0.368	0.368	0.368

Finally, Figure 6.13 shows the set of 15 ranking results for the query "*true grit*" extracted from the interface of the GTE-Rank$_2$ web service over the WCRank_DS1 dataset. The number in red color is the ranking position initially obtained by Bing search engine, i.e. BRank. The values in front of the snippet ID, reflect the ranking value computed by the GTE-Rank methodology, i.e. GRank. It is interesting to note that our algorithm retrieves in the second, third, sixth, ninth and tenth position, five relevant results that were initially retrieved by the Bing search engine in the thirty-first, thirty-fifth, thirty-second, forty-seventh and forty-first position, respectively. Moreover, our algorithm is capable of pushing down the not so relevant first result of Bing search engine to the eleventh position.

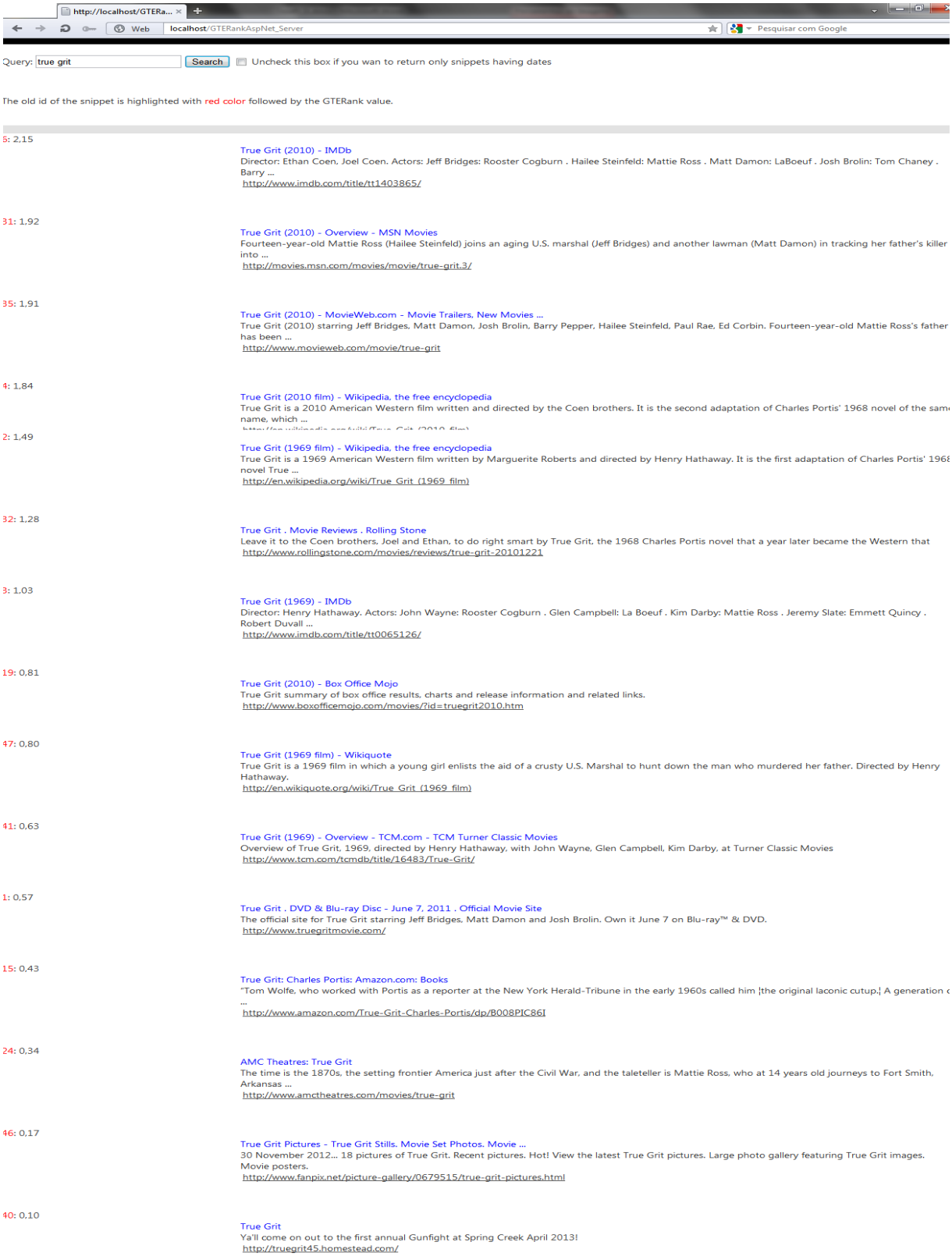


Figure 6.13: GTE-Rank interface for the query "*true grit*" over the WCRank_DS1. Extracted from http://wia.info.unicaen.fr/GTERankAspNet_Server

6.4.3 Experiment C

We now test the performance of GRank on a collection that also includes atemporal web snippets, i.e. texts which do not include any temporal features. In order to do this, we resort to the unreduced WCRank_DS2 dataset and conduct two experiments named **C1** and **C2**. The first one studies the Top approach and the second one the Tail scenario.

Experiment **C1**: Top

In this experiment, we evaluate the GRank performance on the Top approach over the WCRank_DS2 dataset. We start by considering the difference between the GRank algorithm and the baseline methods when varying α from 0.0 to 1.0. An overall analysis of the results (see Table 6.14) show that GRank improves as α increases, which is consistent with the results of Experiment A and B. It is clear however, that this impact is not as evident as in the case of the WCRank_DS1 collection. This is due to the introduction of a set of atemporal texts from the WCRank_DS2 dataset (representing 71.5% of the entire collection), which lowers the importance of the GTE temporal part of the ranking algorithm. For a clearer depiction of this, we recall Equation 6.1 below:

$$GTE\text{-}Rank(q, S_i) = \alpha * \sum_{j=1}^{u} GTE\left(q, d_{j,i}^{Rel}\right) + (1-\alpha) * \sum_{h=1}^{k} IS\left(q, w_{h,i}\right), \alpha \in [0,1]$$

In fact, the value of α does not matter if the snippet itself contains no candidate dates. In such cases the results are simple computed by the IS temporal part of the ranking formula. Despite this fact, one can note that GRank still outperforms all the baseline methods for α between 0.0 and 1.0, both for MAP and MRP. This is clearly depicted in Figure 6.14 and Figure 6.15, where statistical significance (p-value < 0.05) of the results of GRank over each baseline method, using matched paired one-sided t-test, is represented by the absence of a solid marker in each of the three corresponding lines. This lead us to can conclude that the conceptual part of the ranking formula performs itself quite well. One reason for this is the use of the GTE-Class which makes it possible for q and $w_{h,i}$ to be defined as two context vectors consisting of a combination between relevant words and relevant dates, instead of non-relevant ones. We complement this analysis by comparing the effectiveness of GRank against baselines on pushing down non-relevant documents. The results obtained indicate that GRank achieves the best performance. This is clearly depicted in Figure 6.16 and can be explained by the ability of our system to ward off the non-relevant snippets from the top of the list, due to the use of GTE-Class.

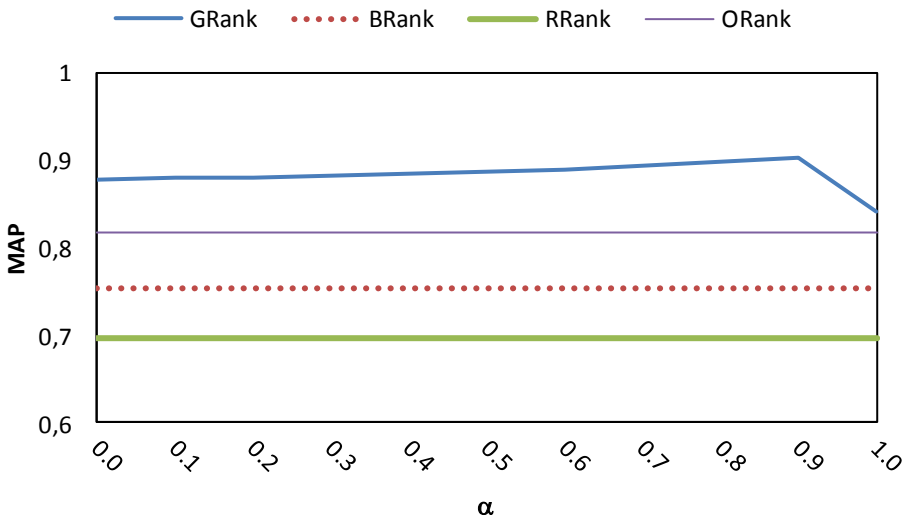


Figure 6.14: MAP. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Top. WCRank_DS2 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test

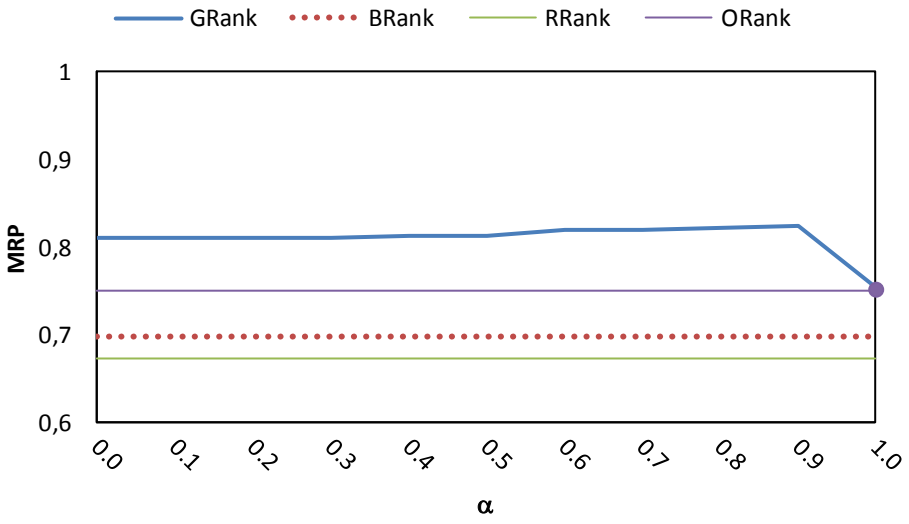


Figure 6.15: MRP. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Top. WCRank_DS2 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test

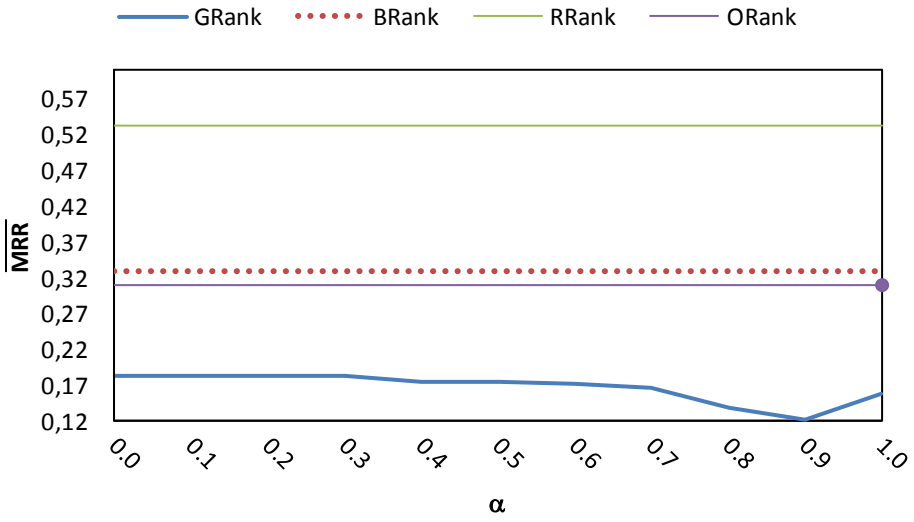


Figure 6.16: $\overline{MRR}$. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Top. WCRank_DS2 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test

A summary of the results is shown in Table 6.14 for the three different metrics, MAP, MRP and $\overline{MRR}$, plus P@k and NDCG@k, registering statistical significance of our algorithm in almost all the cases. Note that the second best approach in this experiment is the ORank baseline. This is not surprising since this method pulls to the top all the web snippets having dates, which, will naturally result in a enhanced performance. Regardless of this, GRank can still significantly outperform ORank by 0.086 in MAP, 0.074 in MRP, 0.187 in $\overline{MRR}$, 0.097 in P@10 and 0.050 in NDCG@5 when $\alpha = 0.9$. We conclude that simply using a system that pushes to the top documents incorporating possible temporal features, may not be sufficient to achieve a good performance as it is subject to a high degree of randomness. On the one hand, some of the documents will still be relevant to the query although not incorporating any temporal feature. On the other hand, there will be some documents which, although including a temporal pattern, may not be as relevant as those that do not include any date at all (e.g. "*Avatar: The Last Airbender Movie Desktop Wallpaper 1280 x 1024 Pixels*").

Table 6.14: MAP, MRP, $\overline{MRR}$, P@1, P@3, P@5, NDCG@5, NDCG@10 and NDCG@20 results. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Top approach. WCRank_DS2 dataset. The absence of underline indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test.

	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
MAP	GRank	0.878	0.879	0.880	0.881	0.883	0.886	0.889	0.894	0.899	0.903	0.842
	BRank	0.754	0.754	0.754	0.754	0.754	0.754	0.754	0.754	0.754	0.754	0.754
	RRank	0.701	0.695	0.705	0.683	0.698	0.694	0.697	0.696	0.680	0.699	0.705
	ORank	0.817	0.817	0.817	0.817	0.817	0.817	0.817	0.817	0.817	0.817	0.817
MRP	GRank	0.809	0.810	0.810	0.810	0.812	0.812	0.818	0.819	0.821	0.824	0.752
	BRank	0.696	0.696	0.696	0.696	0.696	0.696	0.696	0.696	0.696	0.696	0.696
	RRank	0.666	0.671	0.682	0.663	0.670	0.674	0.664	0.681	0.660	0.679	0.676
	ORank	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750
$\overline{MRR}$	GRank	0.182	0.182	0.182	0.183	0.174	0.175	0.171	0.166	0.138	0.122	0.156
	BRank	0.328	0.328	0.328	0.328	0.328	0.328	0.328	0.328	0.328	0.328	0.328
	RRank	0.497	0.498	0.534	0.588	0.486	0.604	0.524	0.548	0.578	0.540	0.467
	ORank	0.309	0.309	0.309	0.309	0.309	0.309	0.309	0.309	0.309	0.309	0.309
P@5	GRank	0.937	0.932	0.932	0.932	0.932	0.932	0.937	0.937	0.958	0.958	0.926
	BRank	0.795	0.795	0.795	0.795	0.795	0.795	0.795	0.795	0.795	0.795	0.795
	RRank	0.695	0.632	0.616	0.653	0.632	0.605	0.737	0.653	0.611	0.663	0.684
	ORank	0.868	0.868	0.868	0.868	0.868	0.868	0.868	0.868	0.868	0.868	0.868
P@10	GRank	0.929	0.929	0.926	0.926	0.929	0.934	0.939	0.939	0.942	0.942	0.887
	BRank	0.784	0.784	0.784	0.784	0.784	0.784	0.784	0.784	0.784	0.784	0.784
	RRank	0.658	0.626	0.634	0.653	0.647	0.655	0.718	0.645	0.642	0.655	0.676
	ORank	0.845	0.845	0.845	0.845	0.845	0.845	0.845	0.845	0.845	0.845	0.845
P@20	GRank	0.876	0.876	0.879	0.880	0.882	0.886	0.884	0.888	0.888	0.878	0.811
	BRank	0.721	0.721	0.721	0.721	0.721	0.721	0.721	0.721	0.721	0.721	0.721
	RRank	0.670	0.651	0.663	0.663	0.664	0.680	0.689	0.657	0.650	0.661	0.663
	ORank	0.797	0.797	0.797	0.797	0.797	0.797	0.797	0.797	0.797	0.797	0.797
NDCG@5	GRank	0.934	0.937	0.940	0.940	0.944	0.946	0.947	0.949	0.974	0.979	0.971
	BRank	0.933	0.933	0.933	0.933	0.933	0.933	0.933	0.933	0.933	0.933	0.933
	RRank	0.838	0.799	0.814	0.801	0.847	0.766	0.847	0.820	0.796	0.796	0.838
	ORank	0.929	0.929	0.929	0.929	0.929	0.929	0.929	0.929	0.929	0.929	0.929
NDCG@10	GRank	0.920	0.922	0.924	0.924	0.929	0.928	0.931	0.939	0.962	0.976	0.969
	BRank	0.899	0.899	0.899	0.899	0.899	0.899	0.899	0.899	0.899	0.899	0.899
	RRank	0.804	0.792	0.791	0.782	0.809	0.765	0.816	0.802	0.753	0.781	0.816
	ORank	0.937	0.937	0.937	0.937	0.937	0.937	0.937	0.937	0.937	0.937	0.937

	Method	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
NDCG@20	GRank	0.912	0.914	0.915	0.915	0.920	0.925	0.930	0.938	0.957	0.971	0.963
	BRank	0.883	0.883	0.883	0.883	0.883	0.883	0.883	0.883	0.883	0.883	0.883
	RRank	0.789	0.780	0.787	0.769	0.789	0.769	0.804	0.788	0.751	0.780	0.798
	ORank	0.937	0.937	0.937	0.937	0.937	0.937	0.937	0.937	0.937	0.937	0.937

Once again, we perform nine independent 5-fold cross validation rounds for the $\overline{MRR}$, MAP, MRP, P@5, P@10, P@20, NDCG@5, NDCG@10 and NDCG@20 metrics. Table 6.15 summarizes all these values for the Top evaluation scenario. A detailed analysis of the table shows that GRank outperforms, with statistical significance, the baselines in almost all cases, proving that GRank is capable of obtaining a good performance even over atemporal texts. Generally speaking, we can conclude that the effectiveness of GRank is maximized when α=0.9. This is in line with the results of Table 6.14, where GRank proves to be statistically significant better than ORank and even BRank, only when α is approximately 0.9.

In what follows, we explore the results of P@k on a per-query basis. For that end, we use average precision histograms for each query (see Figure 6.17), as explained in Section 6.3.3. Finally, Figure 6.18 shows Precision/Recall curves based on the results presented in Table 6.16.

Table 6.15: P@k, NDCG@k, MAP, MRP and $\overline{\text{MRR}}$ results. GRank vs. Baselines. Top approach. WCRank_DS2 dataset. The absence of underline indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test.

Method	P@5	P@10	P@20	NDCG@5	NDCG@10	NDCG@20	MAP	MRP	$\overline{MRR}$
	α = 0.86	α = 0.86	α = 0.78	α = 0.90	α = 0.90	α = 0.90	α = 0.88	α = 0.88	α = 0.88
GRank	0.950	0.938	0.886	0.962	0.975	0.969	0.890	0.812	0.151
BRank	0.795	0.786	0.724	0.932	0.899	0.884	0.750	0.691	0.331
RRank	0.704	0.716	0.694	0.821	0.784	0.785	0.698	0.671	0.509
ORank	0.868	0.845	0.799	0.921	0.929	0.930	0.800	0.741	0.331

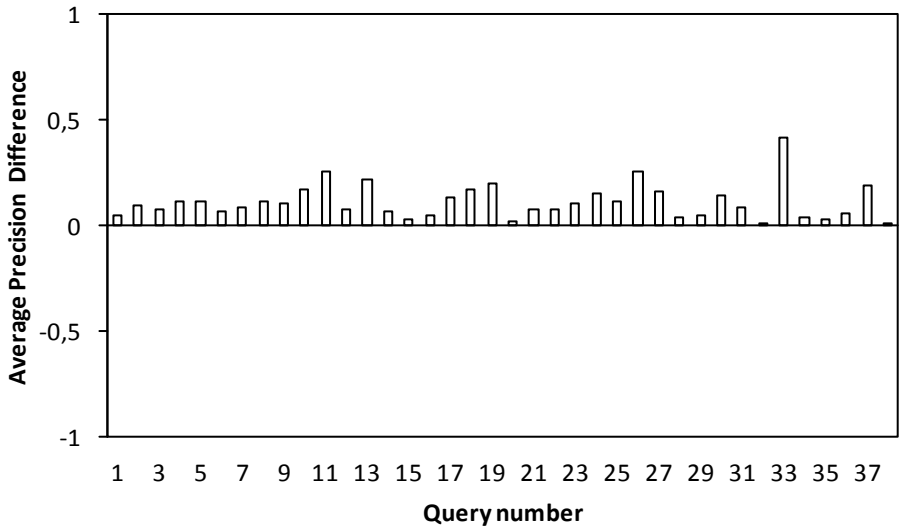


Figure 6.17: Average precision difference histogram for the 38 queries. GRank ($\alpha = 0.9$) vs. Baselines. Top. WCRank_DS2 dataset.

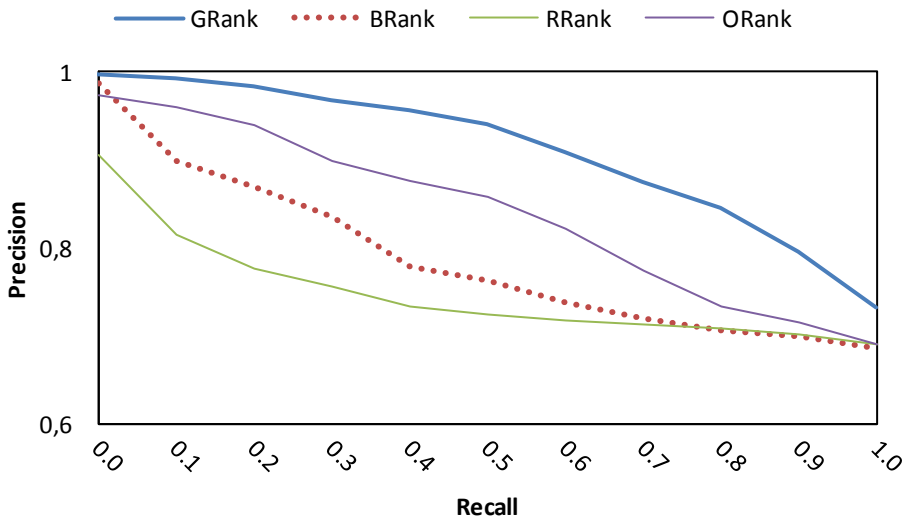


Figure 6.18: Average Recall-Precision for the 38 queries. GRank ($\alpha = 0.9$) vs. Baselines. Top. WCRank_DS2 dataset.

Table 6.16: Precision/Recall curve GRank ($\alpha = 0.9$) vs. Baselines. Top approach. WCRank_DS2 dataset.

Method	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
GRank	0.997	0.992	0.983	0.967	0.956	0.940	0.909	0.876	0.845	0.795	0.732
BRank	0.987	0.899	0.868	0.835	0.779	0.762	0.737	0.720	0.706	0.699	0.686
RRank	0.904	0.814	0.776	0.757	0.733	0.723	0.718	0.713	0.708	0.701	0.689
ORank	0.974	0.960	0.940	0.899	0.877	0.857	0.821	0.773	0.732	0.714	0.690

We conclude this experiment in Figure 6.19 showing the Top 10 ranking results for the query "*true grit*" extracted from the interface of the GTE-Rank$_1$ web service. The number in red color is the ranking position initially obtained by Bing search engine, i.e. BRank. The values in front of the snippet ID, reflect the ranking value computed by the GTE-Rank methodology, i.e. GRank. It is worth noting that similarly to the previous interface discussed, our algorithm is capable of promoting relevant temporal documents to the top, that were initially far down in Bing's search engine list of result's. Furthermore, we show that our algorithm is also able to promote to the top relevant documents, which do not include any temporal expression.

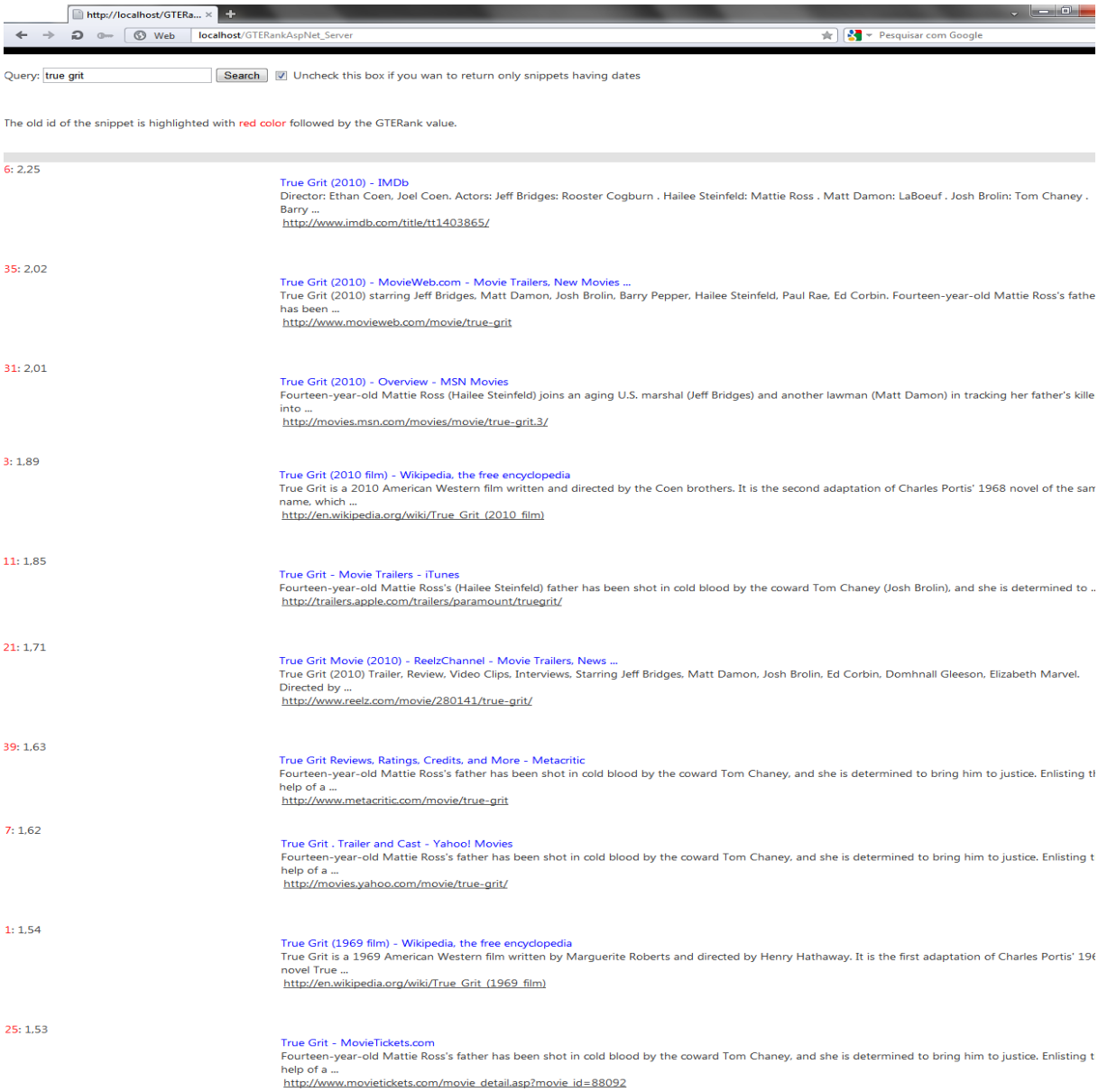


Figure 6.19: Interface of the GTE-Rank web service for the query "*true grit*" over the WCRank_DS2. Top 10 results.

Experiment **C2**: Tail

Finally, in this experiment, we compare the GRank performance against baseline methods over the WCRank_DS2 dataset for the Tail approach. Again, the largest differences are observed in this evaluation scenario. This is shown in Figure 6.20 and Figure 6.21, which demonstrate the ability of GRank to push down non-relevant documents, which were originally ranked higher by BRank. Specifically, we report an increased performance over the BRank baseline of 0.270 for the MAP metric and 0.263 for the MRP one, when $\alpha = 0.9$.

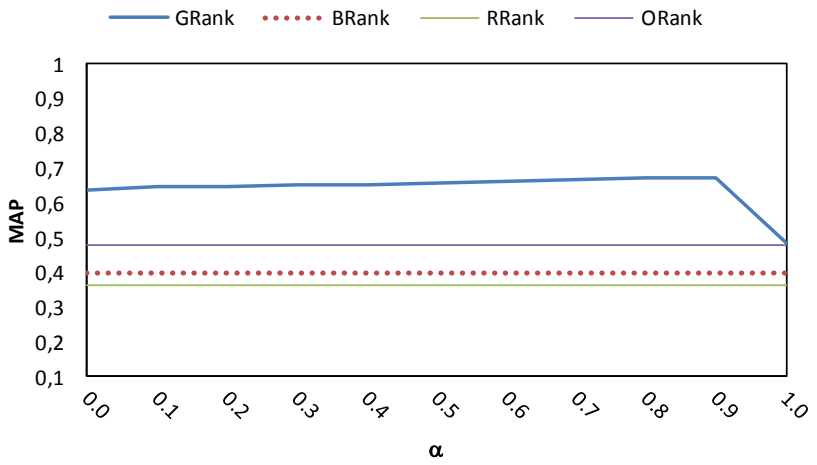


Figure 6.20: MAP. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Tail. WCRank_DS2 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test

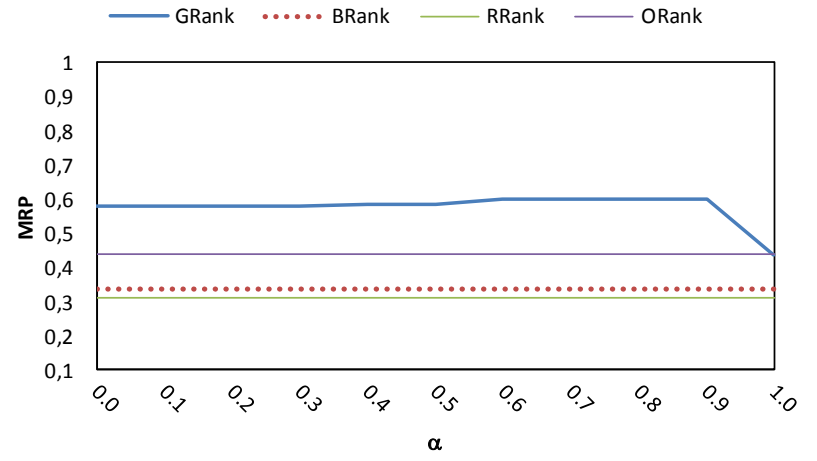


Figure 6.21: MRP. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Tail. WCRank_DS2 dataset. The absence of a solid marker indicates statistical significance of the results of GRank compared with the corresponding baseline methods with p-value < 0.05 using the matched paired one-sided t-test

From these figures we can also conclude that GRank tends to slightly improve as α increases, indicating that the temporal part of our ranking measure has a positive effect in the quality of the retrieved results. This is particularly evident when α varies between 0.0 and 0.9, with GRank being improved in 0.031, 0.019 and 0.021 for MAP, MRP and P@5 respectively. Similarly to experiment B, results get worse when the value of α changes to 1.0. We can therefore conclude that the best results come from the combination between the temporal factor and the conceptual one. A summary of the results is shown in Table 6.17. Plus, a detailed analysis of the table shows that GRank outperforms, with statistical significance, the baselines in all cases for MAP, MRP, P@5, P@10 and P@20 evaluation metrics.

Table 6.17: MAP, MRP, P@5, P@10 and P@20 results. GRank ($0.0 \leq \alpha \leq 1.0$) vs. Baselines. Tail approach. WCRank_DS2 dataset. All the comparisons are statistically significant with p-value < 0.05 using the matched paired one-sided t-test.

	Method	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
MAP	GRank	0.639	0.645	0.648	0.651	0.653	0.658	0.662	0.667	0.670	0.670	0.486
	BRank	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
	RRank	0.357	0.359	0.367	0.351	0.360	0.369	0.344	0.375	0.353	0.397	0.376
	ORank	0.478	0.478	0.478	0.478	0.478	0.478	0.478	0.478	0.478	0.478	0.478
MRP	GRank	0.581	0.580	0.580	0.580	0.586	0.586	0.598	0.600	0.599	0.600	0.437
	BRank	0.337	0.337	0.337	0.337	0.337	0.337	0.337	0.337	0.337	0.337	0.337
	RRank	0.306	0.306	0.317	0.301	0.294	0.308	0.293	0.335	0.297	0.339	0.314
	ORank	0.437	0.437	0.437	0.437	0.437	0.437	0.437	0.437	0.437	0.437	0.437
P@5	GRank	0.705	0.711	0.711	0.721	0.716	0.721	0.721	0.726	0.726	0.726	0.453
	BRank	0.368	0.368	0.368	0.368	0.368	0.368	0.368	0.368	0.368	0.368	0.368
	RRank	0.342	0.284	0.321	0.316	0.321	0.326	0.384	0.326	0.316	0.368	0.353
	ORank	0.442	0.442	0.442	0.442	0.442	0.442	0.442	0.442	0.442	0.442	0.442
P@10	GRank	0.658	0.661	0.664	0.661	0.662	0.667	0.672	0.675	0.672	0.670	0.449
	BRank	0.355	0.355	0.355	0.355	0.355	0.355	0.355	0.355	0.355	0.355	0.355
	RRank	0.339	0.304	0.338	0.300	0.297	0.322	0.378	0.349	0.328	0.334	0.343
	ORank	0.441	0.441	0.441	0.441	0.441	0.441	0.441	0.441	0.441	0.441	0.441
P@20	GRank	0.717	0.714	0.713	0.716	0.714	0.719	0.730	0.730	0.731	0.731	0.586
	BRank	0.452	0.452	0.452	0.452	0.452	0.452	0.452	0.452	0.452	0.452	0.452
	RRank	0.441	0.392	0.423	0.436	0.408	0.447	0.486	0.415	0.408	0.419	0.452
	ORank	0.560	0.560	0.560	0.560	0.560	0.560	0.560	0.560	0.560	0.560	0.560

Once again we perform 5-fold cross validation for the MAP, MRP, P@5, P@10 and P@20 metrics. Results are summarized in Table 6.18 and show that GRank outperforms, with statistical significance, the baselines methods in all the cases, which is consistent with the results observed in WCRank_DS1. In particular, we report an increased performance of GRank over the ORank baseline, of 0.279 for the P@5, 0.223 for the P@10, 0.153 for the P@20 metric, 0.191 for the MAP and 0.148 for the MRP, which demonstrates the problems underlying the ORank baseline method (already described in the scope of the Top evaluation scenario). In addition, the histogram shown in Figure 6.22 demonstrates that there is a significant difference between the average precision of GRank and the three baseline methods. As a matter of fact, GRank performs worse only in a single query.

Table 6.18: P@k, MAP and MRP results. GRank vs. Baselines. Top approach. WCRank_DS2 dataset. All the comparisons are statistically significant with p-value < 0.05 using the matched paired one-sided t-test.

Method	**P@5**	**P@10**	**P@20**	**MAP**	**MRP**
	$\alpha = 0.90$	$\alpha = 0.74$	$\alpha = 0.70$	$\alpha = 0.78$	$\alpha = 0.76$
GRank	0.720	0.665	0.715	0.664	0.585
BRank	0.367	0.353	0.452	0.405	0.347
RRank	0.314	0.315	0.451	0.380	0.318
ORank	0.441	0.442	0.562	0.473	0.437

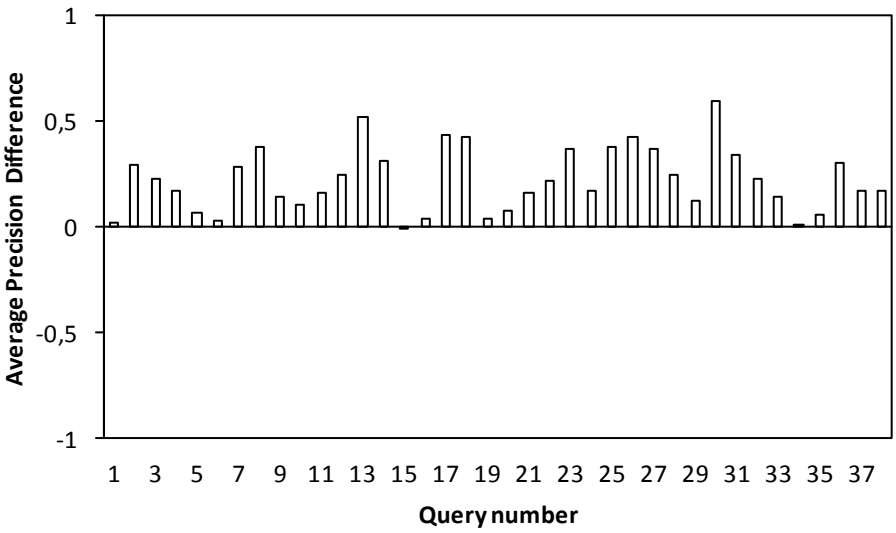


Figure 6.22: Average precision difference histogram for the 38 queries. GRank ($\alpha = 0.8$) vs. Baselines. Tail. WCRank_DS2 dataset.

We provide Precision/Recall curves in Figure 6.23 based on the results presented in Table 6.19. While GRank ($\alpha = 0.8$) performs well for all the recall levels, its performance naturally decreases as it approaches 1.0 of recall. This is particularly observable when moving from 0.5 of recall to 1.0, with a decrease of 0.248, which suggests that while some of the non-relevant documents are still mistakenly dispersed in higher up positions, some of the relevant ones are still incorrectly placed in the lower part of the results. Two reasons for this can be advanced.

Firstly, there are some documents for which a date is not relevant, yet GTE-Class defines it as such, or the opposite, i.e. documents for which a date is relevant, yet GTE-Class defines it as non-relevant. In this regard, it is important to note that the GTE-Class aims to date implicit temporal queries, and not to evaluate the relevance of dates within documents. Thus, it can determine that the date "*2011*" is a relevant year for the query "*Steve Jobs*", but it cannot evaluate whether this date is relevant within a snippet (e.g. "*Steve Jobs - February 24, 1955 – October 5, 2011*") and non-relevant within another one (e.g. "*Steve Jobs fielded some customer service requests updated: Wed Nov 23 2011 05:51:00*"). This issue must clearly be improved in future research.

Secondly, there are some texts, which tend to be pulled up, even if they are not temporally related with the query. This is mostly due to the existence of a few text expressions, which are relevant, not with the query itself, but with some facet of the query. Example 6.1 shows an example of a text retrieved for the query "*Tour Eiffel*", which is relevant for the food stores facet.

> France. The highest rated Food Stores near La **Tour Eiffel** Pastry Shop La **Tour Eiffel** Pastry Shop on *1175* PEMBINA HWY We have bought family birthday cakes here for the past 50...

Example 6.1: Faceted text result of the query "*Tour Eiffel*".

We can note that although 1175 has been correctly detected by the GTE-Class as a non-relevant temporal pattern, still the GTE-Rank algorithm will tend to pull the document up as it includes a few relevant text expressions, more precisely "La Tour Eiffel" and "France". One possible way to overcome this is to apply a temporal clustering approach that is able, not only to detect the temporal issues of the query, but also faceted query topics. This is again another important issue for future work and can be handled with multifaceted state-of-the-art clustering algorithms such as proposed in Scaiella et al. [78]. Notwithstanding the limitations laid out above, GRank can still outperforms the second best approach - i.e. ORank - in 0.092 when the recall level equals to 1.0.

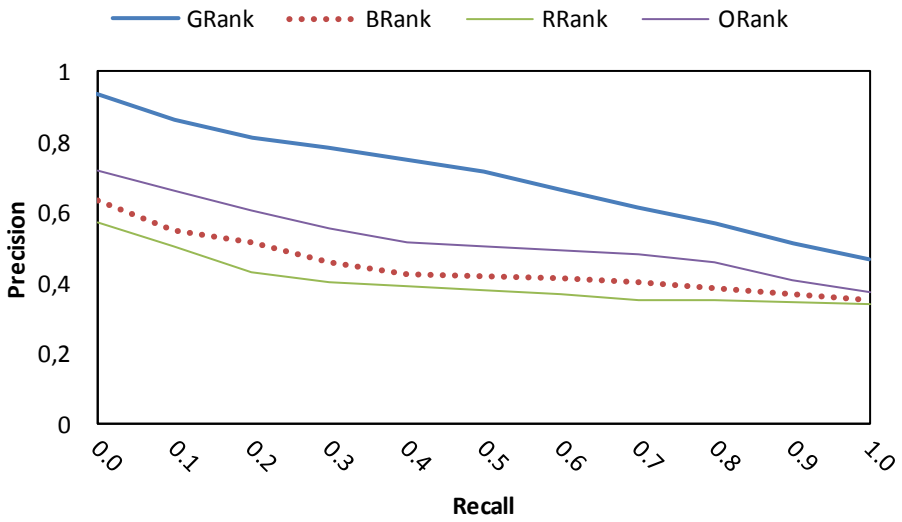
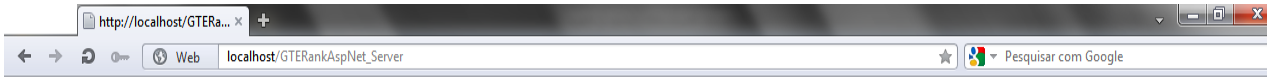


Figure 6.23: Average recall-precision for the 38 queries. GRank ($\alpha = 0.8$) vs. Baselines. Tail. WCRank_DS2 dataset.

Table 6.19: Precision/Recall curve GRank ($\alpha = 0.8$) vs. Baselines. Tail approach. WCRank_DS2 dataset.

Method	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
GRank	0.936	0.864	0.812	0.783	0.751	0.715	0.663	0.616	0.570	0.511	0.467
BRank	0.631	0.547	0.516	0.456	0.422	0.422	0.410	0.400	0.385	0.368	0.352
RRank	0.571	0.506	0.429	0.399	0.390	0.390	0.367	0.351	0.349	0.343	0.337
ORank	0.721	0.660	0.603	0.552	0.516	0.516	0.492	0.478	0.456	0.406	0.375

Finally, Figure 6.24 shows the Tail 10 ranking results for the query "*true grit*" extracted from the interface of the GTE-Rank$_1$ web service (with α set to 0.8) . The number in red color is the ranking position initially obtained by Bing search engine, i.e. BRank. The values in front of the snippet ID, reflect the ranking value computed by the GTE-Rank methodology, i.e. GRank. It is interesting to note that our algorithm is able to position well down in the list, temporally non-relevant documents, that were initially positioned at top positions of the Bing search engine result's, of which IDs numbers 5, 13 and 17 are elucidative examples.



36: 0,20

True Grit - Compare Prices, Reviews and Buy at Nextag
True Grit - 880 results like True Grit Men's Soft Slub Jersey Crew Shirt with Contrast Stitch, True Grit Men's Luxe Fleece Crew Sweatshirt, WTF Wake The Freak Pre ...
http://www.nextag.com/true-grit/products-html#!

13: 0,19

'True Grit' Trailer HD - YouTube
For more info on 'True Grit' visit: http://www.hollywood.com
http://www.youtube.com/watch?v=CUiCu-zuAgM

46: 0,18

True Grit Pictures - True Grit Stills. Movie Set Photos. Movie ...
30 November 2012... 18 pictures of True Grit. Recent pictures. Hot! View the latest True Grit pictures. Large photo gallery featuring True Grit images. Movie posters.
http://www.fanpix.net/picture-gallery/0679515/true-grit-pictures.html

5: 0,10

Welcome to True Grit

http://truegrit.com/

17: 0,10

True Grit - Rotten Tomatoes
In fine Hollywood tradition, John Wayne had to play a "one-eyed fat man" before the Motion Picture Academy considered him worthy of an Oscar. In True Grit, Wayne ...
http://www.rottentomatoes.com/m/true_grit/

23: 0,10

True Grit in Clothing, Shoes & Accessories . eBay
Find great deals on eBay for True Grit and Lady and the Tramp in Clothing, Shoes & Accessories. Shop with confidence.
http://www.ebay.com/sch/i.html?_nkw=True+Grit

26: 0,10

True Grit up to 82% off at Sierra Trading Post
Save 35-70% Every Day on True Grit at Sierra Trading Post! Great Deals. Great Brands.
http://www.sierratradingpost.com/true-grit~b~20538/

37: 0,10

True Grit Clothing
True Grit Clothing. Welcome Guest Sign In: 0 Items in your bag. about us collection my account order status view bag: Home. MEN: WOMEN: Sign up for email updates about ...
http://www.truegritstock.com/

40: 0,10

True Grit
Ya'll come on out to the first annual Gunfight at Spring Creek April 2013!
http://truegrit45.homestead.com/

48: 0,10

True Grit Cody Floral Cardigan Sweater - Burnout Rib-Knit (For ...
True Grit Cody Floral Cardigan Sweater - Burnout Rib-Knit (For Women) at Sierra Trading Post. Great Deals. Great Brands. Styles in Black, Olive, Navy, and more.
http://www.sierratradingpost.com/true-grit-cody-floral-cardigan-sweater-burnout-rib-knit-for-women~p~5355w/

Figure 6.24: GTE-Rank interface for the query "*true grit*" over the WCRank_DS2. Tail 10 results. Extracted from http://wia.info.unicaen.fr/GTERankAspNet_Server

6.5 Summary

In this chapter, we proposed to adjust the score of a document in a ranking task in response to a given implicit temporal query by following a content-based approach that extracts temporal features from the contents of the document. Our aim was to retrieve, in the top list of results, documents that are not only topically relevant but that are also from the most important time periods. This is a very challenging issue since we need not only to return the most relevant documents that meet the users' query intents, but also to simultaneously devalue those incorporating non-relevant concepts or dates. For this purpose, we developed GTE-Rank, a re-ranking algorithm that combines both conceptual and temporal relevancies in a single score. Through extensive experiments, we demonstrated that GTE-Rank is able to achieve better results under several evaluation metrics compared to three different baselines. More specifically, we showed that the introduction of the GTE-Class causes an improvement of the GTE-Rank performance, both in the Top and in the Tail approaches. This is particularly evident for the latter, where our algorithm showed a notorious capacity to push down non-relevant documents when compared with the baseline methods. Moreover, we also showed the behavior of GTE-Rank under two different types of collections: exclusively temporal ones, and a combination of both temporal and atemporal texts. Even though GTE-Rank performs better under exclusively temporal collections, its effectiveness, still gets significantly improved, with respect to the baselines, when atemporal texts are also considered. Notwithstanding, having achieved such performance, GTE-Rank is still limited to work with the relevance of a candidate date in the query context. This can be overcome in future research, by giving the GTE-Class the capability of also determining the relevance of a candidate date in the context of a document.

In the next chapter, we offer an overview of a new challenging topic called Future Information Retrieval.

Chapter 7

Future Information Retrieval

Over the last few years, a huge amount of temporal written information has become widely available on the Internet with the advent of forums, blogs and social networks. This gave rise to a new and very challenging problem called future retrieval, which was first introduced by Baeza-Yates [8]. The purpose of future retrieval is to extract, from web sources, future temporal information that is known in advance, in order to answer queries that combine text of a future temporal nature. Despite the relevance of this topic, there is little research on the use of temporal information features for future search purposes, and the only known temporal analytics engine is Recorded Future. In this chapter[37], we focus more on future research. In particular, we intend to ascertain whether or not we can apply our techniques to improve the way the future is seen. Following this, two challenging issues need to be considered:

1. Do web documents contain enough temporal information for future analysis?
2. Can text classification and clustering be improved on the basis of existing future-related information contained in web documents?

To answer these questions, we have conducted a comprehensive set of experiments. The results obtained show that web documents are a valuable source of future data that can be particularly useful in identifying and understanding the future temporal nature of a given implicit temporal query. To the best of our knowledge, this is the first study based on a comprehensive

[37] This chapter is partially based on the work published at the Enriching Information Retrieval Workshop associated with SIGIR2011 (Dias, Campos & Jorge 2011) and the Lecture Notes in Artificial Intelligence - Progress in Artificial Intelligence associated with EPIA 2011 (Campos et al. 2011a).

future data analysis having web documents as a data source and implicit temporal queries. This chapter is structured as follows. Section 7.1 provides an overview of related research. Section 7.2 describes the experiments performance. Finally, we summarize this chapter in Section 7.3.

7.1 Related Research

Little research has been conducted so far in this area. Still, there are some studies that do focus on this domain. Kira Radinsky et al. [77] for example, use patterns in web search queries to predict whether an event will appear in tomorrow's news. Mishne & Glance [66] predict movie sales through blogger sentiment analysis. Liu et al. [58], focus on the same line of research and attempt to predict sales performance. More concretely, Baeza-Yates [8] was the first to define this problem of F-IR and to introduce a basic method for searching, indexing and ranking documents according to their future features. Each document is represented by a tuple consisting of a time segment and a confidence probability that measures whether the event will actually happen or not in this time segment. Jatowt et al. on the other hand, propose two studies related with this topic [48, 49]. In their first study [48] the authors approach the problem of generating visual summaries of expected future events suggesting two different methods. The first method takes into consideration the bursts in the frequency of past events in order to estimate the probability that it can occur again in the future. The second method uses the K-Means clustering algorithm to cluster documents containing information about the same future-related event. Each document is represented by a set of both content features and its focus time. The inter-document distance is then defined by linearly combining the distances between their documents content features and their documents focus time, as defined in Equation 7.1:

$$Dist\left(d_i, d_j\right) = (1-\beta).TermDist\left(d_i, d_j\right) + \beta.TimeDist\left(d_i, d_j\right). \quad (7.1)$$

In the experiments reported, the best results in terms of precision occur for $\beta = 0.2$. In consequence, it is clear that the impact of future-related features is relatively reduced.

In their second study [49], the authors conduct an exploratory analysis of future-related information supported on the average number of hits obtained in response to the execution of a set of explicit temporal queries on Bing's search engine. The results allow us to conclude that (1) future-related information clearly decreases after a few years, with some occasional peaks; (2) most of the near future-related contents are related to expected international events, and (3)

distant years are mostly linked to predictions and expectations that relate to issues such as the environment and climate change.

In this chapter, we ask whether web snippets are a valuable source of data that can help deduce the future temporal intent of queries that do not specify a year. Unlike Jatowt et al. [49], our analysis is not based on the execution of queries including explicit future temporal expressions, but it is based on implicit ones. Subsequently, restrictions have not been placed on the language, type and topic of the query. Furthermore, this analysis is not based on the number of hits reported by the search engine, but on the detection and manual analysis of future dates that occur within the set of results retrieved. Moreover, in accordance with the work produced by Jatowt et al. [48], the impact of introducing future features on the process of clustering future-related web contents will be studied. However, unlike this research [48], where only 20 queries were used, we resort to a set of 450 queries. In addition, a classification task is performed. More specifically, each text is classified according to three possible genres: informative web snippets, scheduled texts and rumors.

7.2 Results and Discussion

In this section we discuss the results of two experiments. Section 7.2.1 experimentally evaluates the future temporal nature of web documents and the type of information they present. Section 7.2.2 aims to understand whether data features influence the classification and clustering of future-related texts.

7.2.1 Experiment A

Although we cannot know the future, a lot can be deduced about it by mining huge collections of texts such as weblogs and microblogs (e.g. *Twitter*, *Facebook*). Each of these texts can have a different nature. In this research, we introduce three types of future texts: informative texts, texts about scheduled events and rumors:

1. "*Sony Ericsson Yendo Release Postponed for February 2013 Due to Software Issues*". (Informative);
2. "*The 2022 FIFA World Cup will be the 22nd FIFA World Cup, an international football tournament that is scheduled to take place in 2022 in Qatar"*. (Schedule);

3. "*Avatar 2? Arriving in 2013? James Cameron intends to complete his next film, another 3D epic, within three to four years*". (Rumor)

Understanding the future temporal intent of web documents is, a particularly difficult task, which has been mostly supported by a reliable collection of web news articles annotated with a timestamp. Other possible sources are web documents. However, in contrast to web news articles, web documents, especially those from social networks, suffer from the problem of containing a large number of comments, predictions or plans, all expressed by means of rumors. This has even led some authors [48] to question its credibility. But what can apparently seem like a drawback can actually constitute a great opportunity to infer the users' interests. For example, James Cameron may discover that people are interested in another 3D Avatar movie; mobile companies may redirect their core business to the development of mobile applications due to the growth of this industry that is expected to reach an impressive $35 billion by 2014; environmentalists on the other hand may be interested to know that EasyJet plans to cut its CO2 emissions by 50% until 2015.

In this section, we outline a number of issues on future temporal web mining analysis. This includes for example the temporal value of future dates with regard to a given future year, the frequency of occurrence in a near future temporal window, related categories and text genres. In order to conduct our experiments, we rely on the GISFD_DS dataset which is built upon the Q450R100 collection (of the GISQC_DS dataset) and consists of 62.842 web snippets. We recall (see Section 2.3.1) that in order to form the Q450R100 collection we apply our rule-based model on top of each retrieved result. Each temporal expression is then manually checked so as to keep the set of correct dates only. In the following part, we describe the two experiments conducted, referred to as **A1** and **A2**.

Experiment **A1**: Measuring the Future Temporal Nature of Web Documents

To determine the future temporal value of web snippets, we start by defining two basic measures called $FutureDates(q)$ and $NearFuture(q)$.

$FutureDates(q)$ is defined in Equation 7.2 and can be seen as the ratio between the number of future dates retrieved, divided by the total number of dates retrieved for the query q:

$$FutureDates(q) = \frac{\#\ Future\ Dates\ Retrieved}{\#\ Dates\ Retrieved}, \tag{7.2}$$

where a date is considered of future nature, if, independently of the document timestamp, its focus time is superior to the time when the query was executed. Since the queries were executed on December 2010, the set of dates found in this experiment are considered of future nature, if they are superior to 2010.

Based on this we then classify each document as indicative of a *near* or *distant future* purpose. Documents containing a date from 2011 are classified as a *near future* intention, whereas documents incorporating dates later than 2011 are labeled as having a *distant future* nature. $NearFuture(q)$ is then computed as the ratio between the number of near future dates retrieved, divided by the total number of dates retrieved for the query q:

$$NearFuture(q) = \frac{\#\ 2011\ Dates\ Retrieved}{\#\ Future\ Dates\ Retrieved}. \tag{7.3}$$

The average for all the queries is then determined by applying a micro-average scheme. The number of corresponding items returned for a query is added cumulatively to the values calculated for all the previously computed queries. An example of this is given in Equation 7.4 for the $NearFuture$ measure:

$$NearFuture(.) = \frac{\sum_{i=1}^{|Q|} A_i}{\sum_{i=1}^{|Q|} B_i}, \tag{7.4}$$

where $|Q|$ represents the total number of queries executed, A is the total number of documents retrieved with dates from 2011 for the query i, and B is the total number of documents retrieved with future dates for the query i. $FutureDates(.)$ is computed similarly.

The primary conclusion of our study is that unlike conventional T-IR systems, where the amount of temporal information available is relatively significant, in a future retrieval system, values are naturally lower. That is perfectly clear in Table 7.1, where from a total number of 62.842 web snippets retrieved, 5.777 have temporal features and only 508 are of a future nature. This means that 9.2% of the web snippets contain years, but only 0.81% contain future dates. One reason for this, is that people talk more about the past than the future. This makes it difficult to extract large quantities of future temporal information. Nevertheless, it must be noted that the nature of a search in a conventional system, is naturally different from a search in a future retrieval system, in which not much information is needed to meet the objectives.

Table 7.1: Web snippets future temporal value.

Item	# of Items with Dates		Future Dates			Near Future Dates		
			#	Absolute	Relative	#	Absolute	Relative
Title	2058	3.2%	419	0.6%	20.3%	373	0.5%	88.7%
Snippet	5777	9.2%	508	0.8%	8.7%	419	0.6%	82.4%
Url	3512	5.5%	195	0.3%	5.5%	167	0.2%	85.6%

Subsequently, it is important to note that albeit in a reduced scale, 149 queries, from the total number of 450 queries issued, retrieved at least one future date within the snippet item (see Table 7.2), of which 32 had more than one future date. This means that of the 33.1% queries that retrieved a future date in a snippet, 21.4% had more than one future date. Two of these cases are illustrated in the two following sentences: "*Japan plans to establish a robot moon base by 2020 with a landing by 2015*", and "*FIFA denied that the process for the 2018-2022 World Cup was corrupt*".

Table 7.2: Number of queries resulting in the retrieval of web snippets with future dates.

Item	One Future Date		> One Future Date	
Title	113	25.11%	14	12.38%
Snippet	149	33.11%	32	21.47%
Url	75	16.67%	10	13.33%

Furthermore, we study how future dates are distributed along time. We conclude that, regardless of a continuous shortage of future dates as we move forward in the calendar, a great number of references to far distant years are still found. The occurrence of dates is largely predominant in 2011, but consistent until 2013. Thereafter, there are some quite small peaks in 2014 and 2022 that mostly relate to the Football World Cup, which coincides with the results of Jatowt et al. [49]. Overall, the occurrence of future dates is very common in items retrieved in response to queries belonging to the categories of Automotive (e.g. "*dacia duster"*), Finance & Insurance (e.g. "*bank of America"*), Beauty & Personal Care (e.g. "*hairstyles"*), Sports (e.g. "*football"*) and Computer & Electronics (e.g. "*hp"*). A more detailed analysis of each of the three items: titles, snippets and Urls will now be presented.

Titles. On average, more than 90% of the future dates are related to the near future. This information is mostly related to economic forecasts, such as the expected growth of India, or the prediction that 2011 will be a good year to buy property (based on the fact that queries were

executed at the end of 2010). Some other examples are related to IT companies. For example the release date for electronic devices, or sport events. This is illustrated by the following titles:

"*2011 will be best year to buy a home, says BSA*";
"*Experts bet on India growth story in 2011*";
"*Tour de France organizers unveil climb-heavy 2011 route*";
"*Nokia to launch tablet in Q3 2011*".

As we move forward in the calendar, reference years become scarcer such as with scheduled events, including the Football World Cup or rumors relating to environmental issues or company previews:

"*Mobile App Revenue Estimated at $35 Billion by 2014*";
"*Octopus Paul joins England's 2018 World Cup bid*";
"*Qatar Plans 'Island Stadium' For 2022 World Cup*".

Snippets. The occurrence of future dates in web snippets is not very common. In fact, despite 33.11% of the queries (149 out of 450) retrieve at least one future date within the snippet item, only 8.79% of the items retrieved (508 out of 5777) include a future temporal reference. This clearly contrasts with the values occurred in titles, where 20.35% of the items retrieved (419 out of 2058) include a future temporal feature.

Once again, we note that most texts are related to economic forecasts concerning the world crisis. References to upcoming events can also be spotted, such as the Detroit Auto Show and an interesting political text on a visa agreement between Turkey and Azerbaijan:

"*Honda is planning a major jump in hybrid sales in Japan in 2011*";
"*Next-generation Ford 2012 Escape unveiled at the 2011 Detroit Auto Show*";
"*Visa agreement expected to be signed between Turkey and Azerbaijan in 2011*".

As with titles, business plans prevail in far distant years. References to PayPal accounts can be seen, as well as sales of mobile applications or Adidas plans. Even those related to scheduled events have an economic nature, such as the Qatar Football World Cup reference. In addition, there are other quite interesting examples, one related to the translation of the Bible, another to the environment and another with the calendar of holidays until 2070. Some examples include:

"*Avatar 2? in 2013? Cameron intends to complete his next film in 3 to 4 years*";
"*Wycliffe's mission is to see a Bible translation in every language by 2025*";

"*Calendar of all legal Public and Bank Holidays worldwide, until 2070*".

Urls. As expected, the occurrence of future dates in URLs is scarce when compared to snippets or even titles. Indeed, only 5.6% of the links have a future temporal nature. Regardless of the fact that future dates are very uncommon in URLs, they can still be very useful in some specific cases. A careful observation of the list below leads to the conclusion that future dates in URLs are as descriptive as in titles or even in snippets. Predictions are mostly related to IT companies, economic forecasts, and automotives, as this example shows:

"http://www.grist.org/article/2010-11-15-fords-first-electric-car-to-be-sold-in-20-cities-in-2011".

Finally, references to far distant dates also appear in URLs such as:

"*http://msn.foxsports.com/usa-loses-to-qatar-2022-world-cup-bid*".

Experiment **A2**: Text Classification according to the Type of Information

In this second experiment (**A2**) we aim to manually classify each text embodying a future temporal feature with regard to the type of information it refers to. We rely on the set of 419 titles, 508 snippets and 195 URLs embodying future temporal features and we classify them according to three future temporal classes:

- informative texts;
- schedule texts;
- rumor texts.

Each text was manually classified by three annotators. Fleiss' Kappa statistic [42] was used in order to measure the consistency between the different annotators. Results show Kappa was found to be 0.93, meaning an almost perfect agreement between the raters. The results reached show that on average almost 77% (see Table 7.3) of the texts have either an informative nature or concern a scheduled event which has a very high probability of taking place. The remaining 23% relate to rumor texts, which lack confirmation in the future. Some examples are listed below:

"*WebOS tablet will arrive in March 2011. Details are not officially*" (**Rumor**);

"*Tickets for Lady Gaga 2011 Tour*" (**Scheduled Event**);

"*Latest Hairstyles 2011*" (**Informative**).

Table 7.3: Classification of texts according to genre.

Item	# of Items with Future Dates	Scheduled Events		Informative		Rumor	
Title	419	85	20.29%	248	59.19%	86	20.53%
Snippet	508	136	26.77%	255	50.20%	117	23.03%
Url	195	38	19.49%	101	51.79%	56	28.72%

While informative texts mostly occur with near future dates, schedule events and rumor texts occur more frequently with far distant years (see Table 7.4).

Table 7.4: Classification of texts according to genre for near and distant future dates.

	Near Future			Distant Future		
Item	Schedule	Informative	Rumor	Schedule	Informative	Rumor
Title	15.0%	65.4%	19.5%	63.0%	8.7%	28.2%
Snippet	25.7%	55.8%	18.3%	31.4%	23.6%	44.9%
Url	13.7%	56.8%	29.3%	53.5%	21.4%	25.0%

Words such as "*latest*", "*new*", "*review*", "*information*", "*schedule*", "*announce*", "*official*" and "*early*" are usually used to describe the near future in informative texts, such as information on product releases (e.g., "*dacia duster*", "*audi*", "*toyota*", "*ford*", "*honda*", "*nissan*", "*nokia*", "*microsoft*") and upcoming scheduled events (e.g. "*Auto show*"). Figure 7.1 shows a word cloud for near future dates. It was obtained by providing Wordle[38] with a single text resulting from the intersection of title, snippet and Url texts labeled as near future.

Figure 7.1: Word cloud for near future dates.

[38] http://www.wordle.net/ [February 25th, 2013]

As we move forward in the calendar, it is more common for texts to be related to events planned in advance and to also be of a rumor nature. These are associated with events that require confirmation in the future, as shown in Table 7.4. Long term schedule events such as the FIFA Football World Cup in Brazil and also in Qatar, and rumor words such as "*planning*", "*report*", "*preview*", "*coming*", "*expecting*", "*rumor*", "*scenarios"*, "*reveal*" and "*around*" often replace words with a near future nature, such as "*early*" or "*new*". Figure 7.2 shows a word cloud for distant future dates, following the same procedure laid out above.

Figure 7.2: Word cloud for distant future dates.

Another interesting aspect worth highlighting is that future dates are mostly year related and fewer are related to months or days. This becomes more evident as we move further into the future. Exceptions only occur with scheduled events. The following sentence is an illustrative example: "*Tour de France: from Saturday July 2nd to Sunday July 24th 2011, the 98th*".

7.2.2 Experiment B

In this experiment, we aim to understand whether data features influence the classification and clustering of future-related texts according to their nature: informative, scheduled or rumor. It is important to note that our goal here is not to achieve high accuracy results, but to understand if these three genres can be discovered by simply using specific linguistic features, thus avoiding the importance of time for these tasks, or if instead, the inclusion of temporal features plays an important role. In order to reach a conclusion we conduct two experiments, called **B1** and **B2**.

Experiment **B1**: Classification of Future-Related Texts

This experiment (**B1**) includes cross-domain experiments by selecting and issuing queries for the set of 27 categories available from the Q450R100 collection. The Aue & Gamon [7] and Boey et

al. [12] model that suggests training a classifier on a mixed-domain set, in order to tackle cross-domain learning, was used. Experiments are based on two collections: one consisting of 508 snippets and another consisting of 419 text titles, both tagged with future dates. Url texts were not included in this experiment. From these two collections we end up selecting a balanced number of the different type of future-related texts. Therefore, from the first set of 508 snippets, we end up selecting 117 of Informative nature, 117 of Scheduled intent and 117 of Rumor purpose. From the second collection, which consists of 419 text titles, we collect 86 texts of Informative nature, 86 of Scheduled intent and 86 of Rumor purpose. The final result is a set of 351 balanced texts snippets and 258 balanced text titles, from which four datasets D1, D2, D3 and D4 (see Table 7.5) were built, respectively. Each dataset is labeled with the respective text genre/class. In particular, (D1) consists of texts containing years, (D2) consists of texts withdrawing their years, (D3) consists of texts formed by years plus the mention of their belonging to a near or distant future and (D4) consists of texts without years plus the mention of their belonging to a near/distant future.

Table 7.5: Datasets structure.

Dataset	Web Snippet		Near/Distant Future	Class
	Unigram	Year Dates		
D1	x	x		x
D2	x			x
D3	x	x	x	x
D4	x		x	x

Experiments are run on the basis of a stratified *5*-fold cross-validation for boolean and tf-idf unigram features for five different classifiers:

- Naive Bayes algorithm (boolean);
- K-NN (k = 10, boolean);
- Multinomial Naive Bayes algorithm (tf-idf);
- Weighted K-NN (K = 10 and weight=1/distance, tf-idf);
- Multi-Class SVM (boolean and tf-idf).

Results are presented in Table 7.6 and show that the importance of temporal features in the classification task is heterogeneous, as it depends on the learning algorithm and on text representation. On these grounds, we may conclude that hypothesis **H7** which states "*Temporal*

features detected in web documents improve the predictive ability of correctly classifying future-related texts into one of the three following categories: informative, scheduled or rumor" cannot be verified in all cases.

Table 7.6: Snippet classification results for the boolean and tf-idf cases.

Algorithm	Case	Dataset	Accuracy	Scheduled		Informative		Rumor	
				Precision	F-Measure	Precision	F-Measure	Precision	F-Measure
Naïve Bayes	Boolean	D1	78.1%	84.2%	75.4%	77.8%	77.8%	74.1%	80.5%
	Boolean	D2	77.2%	80.8%	74.1%	78.6%	78.6%	73.3%	78.6%
K-NN	Boolean	D1	58.1%	52.0%	58.1%	56.7%	51.4%	67.9%	64.6%
	Boolean	D2	57.0%	48.2%	60.3%	67.3%	43.0%	68.3%	63.3%
Multi-Class SVM	Boolean	D1	79.2%	87,3%	81,3%	75,2%	77,7%	76,6%	78,8%
	Boolean	D2	79.8%	87,0%	80,2%	75,6%	78,7%	78,2%	80,5%
Multi-Class SVM	TF-IDF	D1	75.2%	83,0%	76,5%	69,5%	72,7%	74,8%	76,7%
	TF-IDF	D2	74.4%	85,6%	77,6%	66,9%	71,2%	73,6%	74,8%
M. Naïve Bayes	TF-IDF	D1	76.4%	78.6%	78.6%	79.4%	72.0%	72.3%	78.0%
	TF-IDF	D2	75.8%	76.0%	77.3%	79.6%	72.6%	72.7%	77.1%
Weighted K-NN	TF-IDF	D1	59.3%	87.5%	61.9%	65.3%	49.7%	48.8%	63.3%
	TF-IDF	D2	51.0%	51.5%	55.0%	66.7%	35.2%	46.9%	56.2%
Naïve Bayes	Boolean	D3	78.6%	84.4%	76.1%	77.8%	77.8%	73.9%	80.0%
	Boolean	D4	78.1%	83.5%	75.7%	79.1%	78.4%	73.4%	79.7%
K-NN	Boolean	D3	62.7%	59.1%	62.7%	57.1%	57.6%	74.0%	68.2%
	Boolean	D4	57.6%	50.0%	59.5%	60.7%	50.0%	59.7%	57.2%
Multi-Class SVM	Boolean	D3	78.6%	86,3%	80,4%	73,8%	76,5%	77,2%	79,2%
	Boolean	D4	79.2%	87,1%	80,7%	74,2%	77,6%	77,9%	79,5%
Multi-Class SVM	TF-IDF	D3	74.9%	83.7%	76.3%	67.7%	72,0%	75,8%	76,8%
	TF-IDF	D4	79.2%	87.1%	80.7%	74.2%	77,6%	77,9%	79,5%
M. Naïve Bayes	TF-IDF	D3	75.5%	78.3%	77.6%	78.4%	71.0%	71.2%	77.3%
	TF-IDF	D4	76.5%	75.2%	75.2%	82.8%	73.3%	77.0%	76.1%
Weighted K-NN	TF-IDF	D3	56.4%	86.8%	54.1%	66.7%	49.5%	46.3%	61.3%
	TF-IDF	D4	57.5%	50.0%	59.5%	60.7%	50.7%	68.4%	61.3%

In general, all of the algorithms (see Figure 7.3), with the exception of SVM (boolean) show improved results in terms of accuracy with the simple use of explicit years. The greatest difference is in the Weighted K-NN algorithm. However, both Naïve Bayes and SVM (boolean) largely outperform the Weighted K-NN in terms of accuracy. In contrast, the dates do not have a

great impact if combined with near/distant future knowledge. Indeed, Multi-Class SVM (boolean and tf-idf), Multinomial Naïve Bayes and Weighted K-NN provide better results for D4 than D3. Equally, in the comparison between D1 and D2, the greatest difference in accuracy occurs with the K-NN algorithm. Once again, the Naïve Bayes and SVM (boolean) achieve the best results.

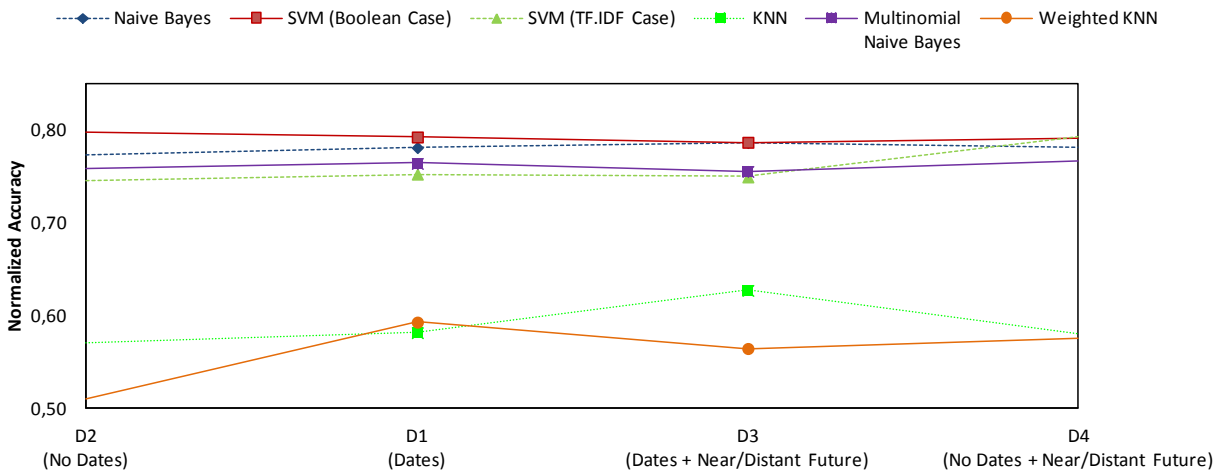


Figure 7.3: Overall analysis of global accuracy for snippet texts.

An individual analysis of each text genre (informative, scheduled, rumor) also led to the conclusion that the introduction of temporal features has an overall positive impact on precision in the classification of scheduled texts. In contrast, the classification of informative texts is more accurate without dates and this is uncertain in the case of rumor texts. Overall these conclusions are confirmed by F-Measure for scheduled and informative texts, but interestingly, not for rumor texts, which show an overall positive impact with F-Measure with the introduction of time features. The best results, however, occur for the SVM algorithm (boolean) without the use of any temporal features. Figure 7.4 shows the results for the specific case of Naïve Bayes.

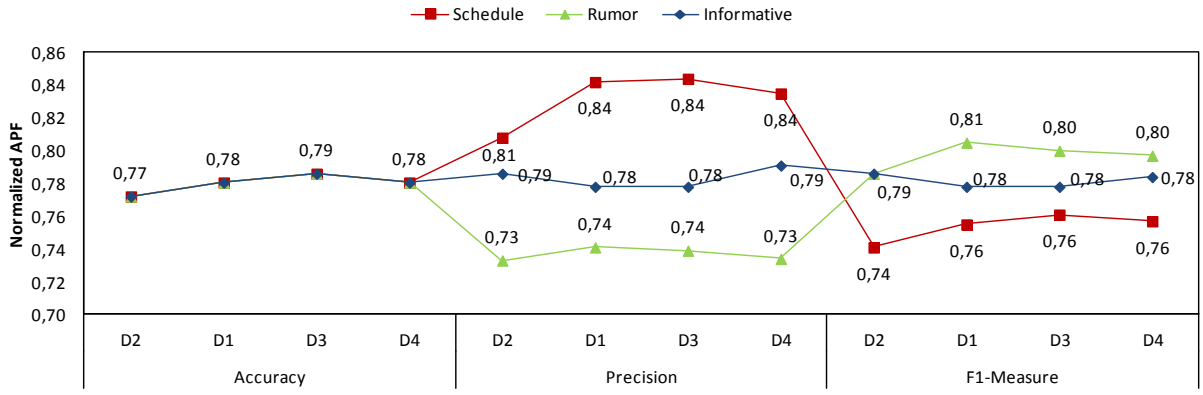


Figure 7.4: Text genre analysis for Naïve Bayes (D1,D2) and (D3,D4) comparison.

The same experiments performed on the web snippets were then performed on the set of 258 balanced text titles. The results are shown in Table 7.7.

Table 7.7: Title classification results for the boolean and tf-idf cases.

Algorithm	Case	Dataset	Accuracy	Scheduled		Informative		Rumor	
				Precision	F-Measure	Precision	F-Measure	Precision	F-Measure
Naïve Bayes	Boolean	D1	78.1%	83.5%	75.7%	79.1%	78.4%	73.4%	79.7%
	Boolean	D2	79.9%	77.8%	83.2%	74.1%	80.0%	96.4%	75.2%
K-NN	Boolean	D1	54.3%	71.6%	62.7%	44.7%	60.4%	100%	24.5%
	Boolean	D2	55.4%	56.9%	67.0%	51.2%	61.0%	100%	17,0%
Multi-Class SVM	Boolean	D1	74,4%	75,0%	75,9%	66,7%	72,3%	85,3%	75,3%
	Boolean	D2	76,4%	74,7%	78,5%	70,5%	74,0%	86,8%	76,6%
Multi-Class SVM	TF-IDF	D1	72.9%	71.4%	73.4%	66.7%	70.3%	83.1%	75.2%
	TF-IDF	D2	76.4%	73.5%	78.3%	71.3%	74.4%	87.9%	76.3%
M. Naïve Bayes	TF-IDF	D1	77.9%	78.9%	80.7%	70.4%	78.4%	90.0%	74.0%
	TF-IDF	D2	76.4%	76.5%	81.5%	69.3%	74.9%	88.1%	71.7%
Weighted K-NN	TF-IDF	D1	53.1%	70.0%	62.8%	43.8%	59.1%	100%	20,8%
	TF.IDF	D2	53.1%	53.5%	64.2%	50.8%	60.0%	100%	11,0%
Naïve Bayes	Boolean	D3	72.9%	71,8%	71,3%	63,6%	74,4%	96,2%	72,5%
	Boolean	D4	77.9%	75,3%	79,8%	71,0%	78,8%	96,3%	74,3%
K-NN	Boolean	D3	53.9%	71,9%	61,3%	44,5%	60,4%	100%	24,5%
	Boolean	D4	52.7%	70,4%	63,7%	43,6%	58,9%	100%	17,0%
Multi-Class SVM	Boolean	D3	75,2%	75,9%	76,3%	67,3%	73,7%	86,6%	75,8%
	Boolean	D4	75,6%	76,4%	77,7%	66,7%	73,3%	89,1%	76,0%
Multi-Class SVM	TF-IDF	D3	73,6%	73.0%	74.3%	67.0%	73.0%	84.8%	73.7%
	TF-IDF	D4	74.4%	75.0%	75.9%	65.7%	73.2%	88.7%	74.3%
M. Naïve Bayes	TF-IDF	D3	77.1%	77.8%	79.5%	70.0%	78.6%	89.7%	72.2%
	TF-IDF	D4	77.1%	76.5%	81.5%	71.3%	77.0%	88.1%	71.1%
Weighted K-NN	TF-IDF	D3	52.3%	69.7%	60,5%	43,4%	59,0%	100%	46,8%
	TF-IDF	D4	51.1%	62.7%	61,5%	44,1%	58,6%	100%	43,7%

Overall, it is clear that most of the algorithms (see Figure 7.5) perform worst in terms of accuracy with the introduction of temporal features, indicating that time characteristics do not have a great impact on the classification task. This does not happen with the Multinomial Naïve Bayes, which has one of the best overall results, only exceeded by the Naïve Bayes algorithm.

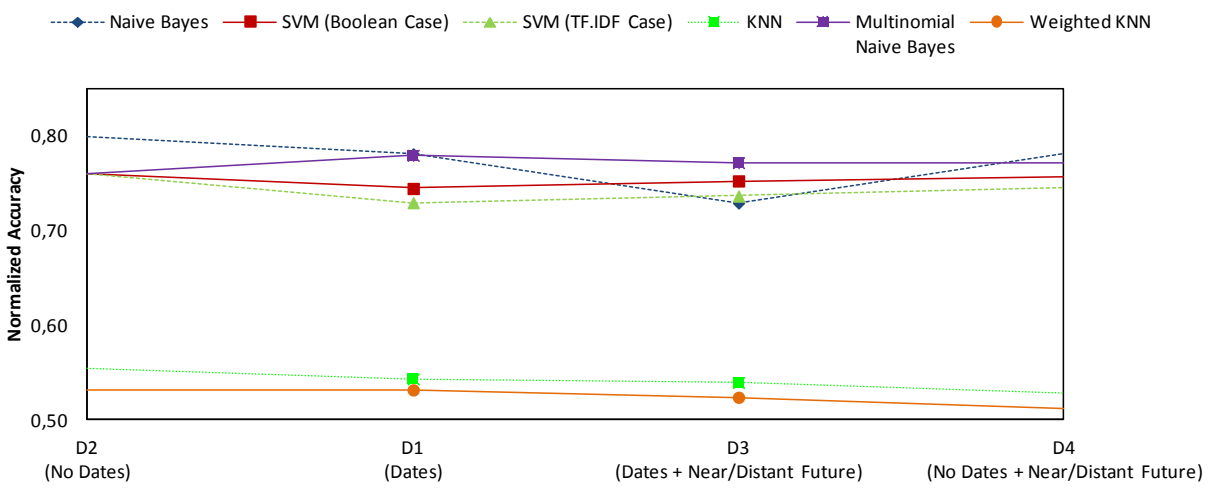


Figure 7.5: Overall analysis of global accuracy for title texts.

This is confirmed by a detailed analysis of all three types of text genres, where the Multinomial Naïve Bayes algorithm shows successful results. Overall, for almost all of the algorithms, scheduled texts benefit from the introduction of temporal features, which is not as clear in the case of informative texts. Another interesting result is that precision in rumor texts is very high. However, with the exception of the Multinomial Naïve Bayes algorithm, time features do not have an overall impact on the classification task. The following figure (see Figure 7.6) shows these results for the specific case of the Multinomial Naïve Bayes algorithm.

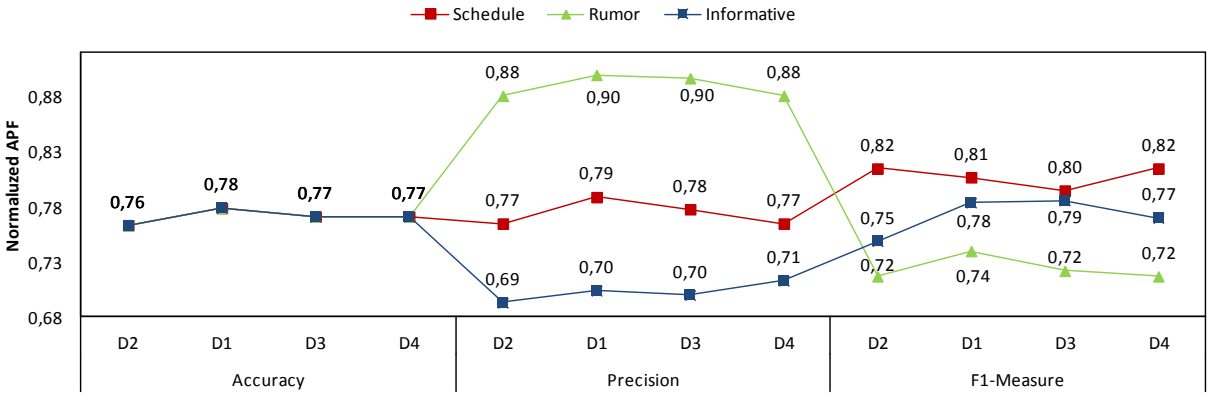


Figure 7.6: Text genre analysis for Multinomial Naïve Bayes (D1,D2) and (D3,D4) comparison.

Experiment **B2**: Clustering of Future-Related Texts

Finally, a set of experiments using the well known K-means clustering algorithm was proposed in order to understand the impact of temporal features within this process. The idea is to automatically retrieve three different clusters (informative, scheduled and rumors) based on the same representations, D1, D2, D3 and D4. As in the classification case, experiments for the boolean and tf-idf cases, and for snippets and text titles are shown.

Results for text snippets are presented in Table 7.8 and show that they are more sensitive to the near/distant future feature, as the best results, for the Boolean case, are obtained for D3. However, the best overall results are obtained by using the K-means over D4, which only takes into account a coarse-grained temporal feature. It must also be noted that scheduled texts have a very high precision rate of almost 85% with a positive impact on the use of temporal features.

Table 7.8: Snippet clustering results for the K-means in the boolean and tf-idf cases.

Algorithm	**Case**	**Dataset**	**Correctly Clustered**	**Scheduled**	**Informative**	**Rumor**
				Precision	**Precision**	**Precision**
K-Means	Boolean	D1	43.59%	34.7%	59.5%	41.1%
		D2	43.59%	34.7%	59.5%	41.1%
		D3	45.02%	36.0%	55.8%	50.0%
		D4	41.88%	33.9%	46.6%	43.6%
	tf-idf	D1	39.04%	84.6%	35.6%	20.0%
		D2	35.90%	83.3%	34.4%	29.4%
		D3	40.74%	25.0%	38.0%	50.6%
		D4	51.00%	43.4%	50.5%	58.4%

This is a clear contrast to text titles clustering, as the best results occur for D3 in the tf-idf representation, with nearly a 13% impact when compared to D4 (Table 7.9). Moreover, the use of temporal features, either alone or combined with near/distant future knowledge, show a positive impact in the clustering task, but for rumor texts they reach an impressive value of almost 85% in terms of precision. The results obtained on this occasion were not conclusive for D1 and D3 (Boolean case), in that more than two clusters were not found. A more detailed analysis led to the conclusion that this is mostly because the system appears to have some difficulties in splitting schedule texts from those of a rumor nature. Similarly to the previous experiment, we may conclude that hypothesis **H8** which states "*Temporal features improve the clustering precision of texts containing references to future events*" cannot be verified in all the cases.

Table 7.9: Title clustering results for the K-means in the boolean and tf-idf cases.

Algorithm	Case	Dataset	Correctly Clustered	Scheduled	Informative	Rumor
				Precision	Precision	Precision
K-Means	Boolean	D1	39,54%			
		D2	42,25%	34.9%	47.5%	84.5%
		D3	39,54%			
		D4	42,25%	34.9%	47.5%	84.5%
	tf-idf	D1	41,87%	34.7%	37.6%	82.4%
		D2	41,87%	37.1%	37.0%	79.3%
		D3	53,49%	68.0%	45.0%	82.8%
		D4	41,87%	37.5%	35.8%	79.3%

7.3 Summary

In this chapter, we conducted an exploratory analysis of future information on the Internet. Results show that titles, particularly in the near future, contain a broad range of temporal information, which is still significant in the case of text snippets and Urls. In addition, we conclude that texts are more often of a scheduled and rumor nature as we move forward in the calendar, contrary to what happens with informative texts, which are unlikely to appear. The high precision of these results and the work presented by Adam Jatowt et al. [48], who has shown that temporal features can help cluster future-related web snippets, led to our final experiments. We performed a set of exhaustive classification and clustering tests based on the three different future-related text genres (informative, scheduled and rumors). The results of our analysis are subject to discussion. Indeed, depending on the representation of the text and on the algorithm family, the temporal issue may or may not have any influence on the classification and clustering of existing future related information.

For the classification task, the SVM and the Naïve Bayes provide the best overall results for text snippets and text titles respectively. However, none of these results was obtained using temporal features. Moreover, the probabilistic learning and the lazy learning families always show the best results for the classification of text snippets when any time feature is used, with the exception of the Multinomial Naive Bayes and the Weighted K-NN for D3. This is the opposite of what happens with the classification of text titles, where most of the algorithms perform better without temporal features. Furthermore, we can also conclude that in general, the introduction of

temporal features has an overall positive impact on the classification of scheduled texts, both in snippets as well as in text titles. Interestingly we can also note that the detection of rumor texts benefits from the introduction of temporal features, particularly in the probabilistic algorithms. For the clustering task, and in particular for the K-means algorithm, the impact of temporal features is more apparent in D1 for snippets and in D3 for text titles. Moreover, the identification of schedule texts is particularly easy in text snippets, while rumor texts are easily identified in text titles.

The results obtained in this emerging IR problem are promising. We believe that this information will serve to improve temporal knowledge in terms of the aims of the user's query, and is a step towards the formation of a future search engine, where the returned documents relate to future periods of time. As such, time features must definitely be treated in a special way and further experiments must be carried out with different representations of time-related features in the learning process, so that more definitive conclusions can be reached.

Chapter 8

Conclusions and Future Research

Despite the fact that web documents contain many temporal expressions, few studies have fully used this information to improve web search diversity. Indeed, most of the IR systems do not yet incorporate temporal features in their architectures, treating all queries as if they were (temporally) equal. This limitation is due to the fact that retrieval models employed continue to represent documents and queries rather simplistically, ignoring their underlying temporal semantics. Subsequently, they fail to understand the users' temporal intents.

The goal of this thesis was to design a model that tackles the temporal dimension of the user's queries, in order to identify not only relevant documents but also relevant time periods. This demands not only the development of better document representations, which include temporal features, but also better temporal similarity metrics capable of reflecting the existing relation between the query and the set of extracted dates.

In order to achieve this, we developed a new temporal similarity measure upon which we studied two classical IR tasks: Clustering and Re-ranking. In particular, we proposed:

- A first study towards a comprehensive temporal analysis of web snippets;
- A simple temporal classification model, capable of determining whether a query is temporal or atemporal, on the basis of web snippets;
- A temporal second-order similarity measure, denoted GTE, which evaluates the degree of relation between candidate dates and a given query;

- A classification methodology (threshold-based), called GTE-Class, which is able to identify the set of top relevant dates for a given implicit temporal query, while filtering out the non-relevant ones;
- A temporal clustering algorithm, named GTE-Cluster, that disambiguates a query with respect to its temporal nature and allows for better browsing;
- A temporal ranking function, called GTE-Rank, that re-ranks web search results based on their temporal intents.

Each of these proposals was experimentally evaluated over a set of real-world text queries. Specifically, we compared the performance of our approach against different proposals under several distinct evaluation metrics. The results obtained showed that our approach is capable of improving the results compared with the different baselines. Both the datasets and the experimental results are available online so that the research community can assess our results and propose new improvements to our methodologies. Furthermore, we made publicly available a set of web services, so that our approach can easily be tested online. Although efficiency was not a core part of the framework, all the solutions perform quite well. This makes our approach an interesting solution to other applications with temporal demands.

Finally, with our eyes set upon the future, we developed a study to ascertain whether the techniques developed in this thesis could be applied to improve the way the future is seen. In particular, we studied the future temporal value of web documents and concluded that web snippets are a rich source that can be used to infer information with a future outlook.

8.1 Future Research

One main limitation of this research is that web snippets are computed by search engines, which we do not control. As a consequence, basing our system upon results generated by a black box may prevent us from obtaining a clear picture of the temporal value of web snippets. In this sense, we aim to evaluate the feasibility of developing a search engine, albeit on small scale, so that this limitation can be overcome.

Furthermore, a new similarity measure that focuses on identifying top relevant dates within a single snippet, in line with what has been proposed by Strötgen et al. [85], can be further

studied. This will contrast with GTE which is particularly focused on retrieving a set of relevant dates to a given query upon processing all web snippets.

While we already achieved the initial stage of flat temporal clustering, our proposal still lacks an approach focused on the topical dimension, so as to ensure that the set of snippets found in the same cluster are query topic-related. As future research, we aim to provide an effective clustering algorithm that clusters and ranks snippets, both based on their temporal and conceptual proximities. For this, we may compute the similarity between two snippets, S_i and S_j, subject to the combination of three different dimensions: (1) Conceptual; (2) Temporal; and (3) Conceptual/Temporal as defined in Equation 8.1:

$$Sim(S_i, S_j) = \left(\lambda \times \sum_{w_1 \in S_i^c} \sum_{w_2 \in S_j^c} Sim_C(w_1, w_2)\right) + \left(\beta \times \sum_{d_1 \in S_i^t} \sum_{d_2 \in S_j^t} Sim_T(d_1, d_2)\right) + \left(\varphi \times \sum_{wd_1 \in S_i^{ct}} \sum_{wd_2 \in S_j^{ct}} Sim_{CT}(wd_1, wd_2)\right), \quad \lambda + \beta + \varphi = 1 \qquad (8.1)$$

where, *C* means Conceptual, *T* Temporal, *CT* Conceptual/Temporal and $Sim_k, k = \{C, T, CT\}$ is any similarity measure (e.g. IS) that computes the similarity between two snippets supported by the M_{CT}^{Rel} matrix, which gathers all the possible "*word*"-"*word*", "*date*"-"*date*" and "*word*"-"*date*" similarities. Based on this new snippet-snippet matrix, one could directly apply any clustering algorithm such as K-means [61] for partitional clustering, Poboc [27] or Clustering by Committee (*CBC*) [71] for soft partitional clustering or even Hierarchical Agglomerative Clustering (*HAC*).

Next, we plan to apply an inter-cluster and intra-cluster ranking algorithm in order to minimize the user effort when looking for relevant results. A straightforward option would be to consider GTE as a means of ranking the temporal clusters, while applying Cluster-Rank (see Equation 8.2) as a way of ranking the snippets inside each cluster C_j. Likewise Equation 6.1, Cluster-Rank could be defined as a ranking algorithm where the estimation level of membership for each snippet S_i found within each cluster C_j would be given by GTE and IS as follows:

$$Cluster\text{-}Rank(C_j, S_i) = \alpha * \sum_{j=1}^{u} GTE(q, d_{j,i}^{Rel}) + (1-\alpha) * \sum_{h=1}^{k} IS(q, w_{h,i}), \alpha \in [0,1] \qquad (8.2)$$

where q is the query, $d_{j,i}^{Rel}$ a relevant date and $w_{h,i}$ is any word of snippet S_i. Similarly, we could apply classical IR ranking metrics to assess the ranking performance upon two different

approaches: an external approach for the inter-cluster ranking and an internal approach for the intra-cluster one.

In addition, we may evaluate the feasibility of integrating within a cluster, a web snippet that has no temporal information, thereby allowing for the retrieval of documents for a given period, even though their contents have no date in them.

Our temporal similarity approach may also serve as the basis for further improvements in several other applications. For example, we may use it to discover what the future will bring. In this respect, we may focus on texts of rumor nature, which as shown in chapter 7, embody some very interesting characteristics. However, instead of just using web snippets, we may also consider the possibility to use twitter posts, which we believe form a very interesting source of future-related events. A possible extension is to track how the opinion of a person, for example politicians, change over time, with regard to some specific topic, from past revelations to future intents. Another aspect, related to query expansion and advertising, is to assess the temporal similarity between any possible queries using some form of temporal correlation, on the assumption that two queries are semantically related if they are temporally related. In this regard Radinski et al. [76], has presented a study where the temporal correlation of words, instead of queries, is measured through a representative time series of its frequency in New York Times articles. Furthermore, we believe that to detect the period of time a topic is related with, constitutes a promising direction of future research. While GTE already detects the possible time span of the query through the lower and upper bound of the determined time, it fails to detect the corresponding sub-periods of the several possible query facets. Such a mechanism, would enable us to offer the user related query period temporal suggestions. For example, the query "*Obama*" would possibly suggest the temporal queries "*Obama 1961 - 2003*", "*Obama Illinois senate member 1997 - 2004*", "*Obama president 2008 - 2012*" or "*Obama president 2012 -* ".

Finally, we intend to intensify our research on temporal image retrieval. Our aim is to help disambiguating any image implicit temporal query with respect to its most important time features and seek to retrieve temporally relevant images. We have recently published a paper on this topic [36].

References

[1] Advanced Research Projects Agency. Software and Intelligent Systems Technology Office. (1993). In MUC-5: Proceedings of the 5th Conference on Message Understanding. Baltimore, Maryland, USA. August 25 – 27.: Morgan Kaufmann Publishers.

[2] Alonso, O., & Gertz, M. (2006). Clustering of Search Results using Temporal Attributes. In SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 597 - 598). Seattle, Washington, USA. August 6 – 11.: ACM Press.

[3] Alonso, O., Baeza-Yates, R., & Gertz, M. (2009). Effectiveness of Temporal Snippets. In WSSP2009: Proceedings of the Workshop on Web Search Result Summarization and Presentation associated with WWW2009: 18th International World Wide Web Conference. Madrid, Spain. April 20 – 24.: ACM Press.

[4] Alonso, O., Gertz, M., & Baeza-Yates, R. (2007). On the Value of Temporal Information in Temporal Information Retrieval. In SIGIR Forum, 41(2), 35 - 41.

[5] Alonso, O., Gertz, M., & Baeza-Yates, R. (2009). Clustering and Exploring Search Results using Timeline Constructions. In CIKM 2009: Proceedings of the 18th International ACM Conference on Information and Knowledge Management. Hong Kong, China. November 2 – 6.: ACM Press.

[6] Alonso, O., Gertz, M., & Baeza-Yates, R. (2011). Enhancing Document Snippets Using Temporal Information. In R. Grossi, F. Sebastiani, & F. Silvestri (Edits.), Lecture Notes in Computer Science, SPIRE2011: 18th International Symposium on String Processing

and Information Retrieval (Vol. 7024, pp. 26 - 31). Pisa, Italy. October 17 - 21.: Springer Berlin / Heidelberg.

[7] Aue, A., & Gamon, M. (2005). Customizing Sentiment Classifiers to New Domains: a Case Study. In RANLP 2005: Proceedings of the International Conference on Recent Advances in Natural Language Processing. Borovets, Bulgaria. September 21 - 23.

[8] Baeza-Yates, R. (2005). Searching the Future. In S. Dominich, I. Ounis, & J.-Y. Nie (Ed.), MFIR2005: Proceedings of the Mathematical/Formal Methods in Information Retrieval Workshop associated with SIGIR 2005: 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Salvador, Brazil. August 15 – 19.: ACM Press.

[9] Barbetta, P. A., Reis, M. M., & Bornia, A. C. (2004). Estatística para Cursos de Engenharia e Informática. Lisboa.: Atlas.

[10] Berberich, K., Bedathur, S., Alonso, O., & Weikum, G. (2010). A Language Modeling Approach for Temporal Information Needs. In C. Gurrin, Y. He, G. Kazai, U. Kruschwitz, S. Little, T. Roelleke, K. V. Rijsbergen (Eds.), In Lecture Notes in Computer Science - Research and Advanced Technology for Digital Libraries, ECIR 2010: 32nd European Conference on Information Retrieval (Vol. 5993/2010, pp. 13 - 25). Milton Keynes, UK. March 28 – 31.: Springer Berlin / Heidelberg.

[11] Berberich, K., Vazirgiannis, M., & Weikum, G. (2005). Time-Aware Authority Ranking. In IM: Internet Mathematics, 2(3), 301 - 332.

[12] Boey, E., Hens, P., Deschacht, K., & Moens, M.-F. (2007). Automatic Sentiment Analysis of On-Line Text. In ELPUB 2007: Proceedings of the 11th International Conference on Electronic Publishing, (pp. 349 - 360). Vienna, Austria. June 13 - 15.

[13] Bollegala, D., Matsuo, Y., & Ishizuka, M. (2007). Measuring Semantic Similarity between Words Using Web Search Engines. In WWW 2007: Proceedings of the 16th International World Wide Web Conference (pp. 757 - 766). Banff, Alberta, Canada. May 8 – 12.: ACM Press.

[14] Callan, J., & Moffat, A. (December de 2012). Panel on use of Proprietary Data. In ACM SIGIR Forum, 46(2), 10 - 18.

[15] Campos, R. (2011). Using k-top Retrieved Web Snippets to Date Temporal Implicit Queries based on Web Content Analysis. In SIGIR 2011: Proceedings of the 34th

Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (p. 1325). Beijing, China. July 24 - 28.: ACM Press.

[16] Campos, R., Dias, G., & Jorge, A. M. (2009). Disambiguating Web Search Results By Topic and Temporal Clustering: A Proposal. In KDIR 2009: Proceedings of the International Conference on Knowledge Discovery and Information Retrieval (pp. 292 - 296). Funchal - Madeira, Portugal. October 6 - 8.

[17] Campos, R., Dias, G., & Jorge, A. M. (2011a). An Exploratory Study on the impact of Temporal Features on the Classification and Clustering of Future-Related Web Documents. In L. Antunes, & H. S. Pinto (Eds.), Lecture Notes in Artificial Intelligence - Progress in Artificial Intelligence - EPIA 2011: 15th Portuguese Conference on Artificial Intelligence associated with APPIA: Portuguese Association for Artificial Intelligence (Vol. 7026/2011, pp. 581 - 596). Lisboa, Portugal. October 10 – 13.: Springer Berlin / Heidelberg.

[18] Campos, R., Dias, G., & Jorge, A. M. (2011b). What is the Temporal Value of Web Snippets? In TWAW 2011: Proceedings of the 1st International Temporal Web Analytics Workshop associated with WWW2011: 20th International World Wide Web Conference. Hyderabad, India. March 28.: CEUR Workshop Proceedings.

[19] Campos, R., Dias, G., Jorge, A. M., & Nunes, C. (2012a). Enriching Temporal Query Understanding through Date Identification: How to Tag Implicit Temporal Queries? In TWAW 2012: Proceedings of the 2nd International Temporal Web Analytics Workshop associated with WWW2012: 21th International World Wide Web Conference (pp. 41 - 48). Lyon, France. April 17.: ACM Press.

[20] Campos, R., Dias, G., Jorge, A. M., & Nunes, C. (2012b). GTE: A Distributional Second-Order Co-Occurrence Approach to Improve the Identification of Top Relevant Dates. In CIKM 2012: Proceedings of the 21st ACM International Conference on Information and Knowledge Management (pp. 2035 - 2039). Maui, Hawaii. October 29 - November 02.: ACM Press.

[21] Campos, R., Jorge, A. M., & Dias, G. (2011c). Using Web Snippets and Query-logs to Measure Implicit Temporal Intents in Queries. In QRU 2011: Proceedings of the Query Representation and Understanding Workshop associated with SIGIR2011: 34th Annual International ACM SIGIR 2012 Conference on Research and Development in Information Retrieval (pp. 13 - 16). Beijing, China. July 28.

[22] Campos, R., Jorge, A. M., Dias, G., & Nunes, C. (2012c). Disambiguating Implicit Temporal Queries by Clustering Top Relevant Dates in Web Snippets. In WIC 2012: IEEE Main Conference Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (pp. 1 - 8). Macau, China. December 4 – 7.: IEEE Computer Society Press.

[23] Chang, A. X., & Manning, C. D. (2012). SUTIME: A Library for Recognizing and Normalizing Time Expressions. In LREC 2012: Proceedings of the 8th International Conference on Language Resources and Evaluation. Istambul, Turkey. May 23 - 25.

[24] Chinchor, N. A. (1998). In MUC-7: Proceedings of the 7th Conference on Message Understanding Conference. Fairfax, Virginia, USA. April 29 - May 1.

[25] Church, K. W., & Hanks, P. (1990). Word Association Norms Mutual Information and Lexicography. In Computational Linguistics, 16(1), 23 - 29.

[26] Cilibrasi, R. L., & Vitányi, P. M. (2007). The Google Similarity Distance. In IEEE Transactions on Knowledge and Data Engineering, 19(3), 370 - 373.

[27] Cleuziou, G., Martin, L., & Vrain, C. (2004). PoBOC: An Overlapping Clustering Algorithm, Application to Rule-Based Classification and Textual Data. In ECAI04: Proceedings of the 16th European Conference on Artificial Intelligence (pp. 440 - 444). Valencia. Spain. August 23 - 27.

[28] Croft, W. B., Metzler, D., & Strohman, T. (2009). Search Engines: Information Retrieval in Practice. New Jersey.: Addison Wesley.

[29] Cunningham, H., Maynard, D., Bontcheva, K., & Tablan, V. (2002). GATE: A Framework And Graphical Development Environment For Robust NLP Tools And Applications. In ACL2002: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (pp. 168 - 175). Philadelphia, PA, USA. July 6 - 12.: Association for Computational Linguistics.

[30] Dai, N., Shokouhi, M., & Davison, B. D. (2011). Learning to Rank for Freshness and Relevance. In SIGIR 2011: Proceedings of the 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 95 - 104). Beijing, China. July 24 - 28.: ACM Press.

[31] Dakka, W., Gravano, L., & Ipeirotis, P. G. (2008). Answering General Time Sensitive Queries. In CIKM 2008: Proceedings of the 17th International ACM Conference on

Information and Knowledge Management (pp. 1437 - 1438). Napa Valley, California, USA. October 26 – 30.: ACM Press.

[32] Deerwester, S., Dumais, S., Landauer, T., Furnas, G., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. In JASIST: Journal of the American Society for Information Science, 41(6), 391 - 407.

[33] Dias, G., Alves, E., & Lopes, J. (2007). Topic Segmentation Algorithms for Text Summarization and Passage Retrieval: An Exhaustive Evaluation. In AAAI 2007: Proceedings of the 22nd Conference on Artificial Intelligence (pp. 1334 - 1340). Vancouver, Canada. July 22 – 26.: AAAI Press.

[34] Dias, G., Campos, R., & Jorge, A. M. (2011). Future Retrieval: What Does the Future Talk About? In ENIR 2011: Proceedings of the Enriching Information Retrieval Workshop associated with SIGIR2011: 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Beijing, China. July 28.

[35] Dias, G., Cleuziou, G., & Machado, D. (2011). Informative Polythetic Hierarchical Ephemeral Clustering. In WIC 2011: IEEE Main Conference Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (pp. 104 - 111). Lyon, France. August 22 – 27.: IEEE Computer Society Press

[36] Dias, G., Moreno, J. G., Jatowt, A., & Campos, R. (2012). Temporal Web Image Retrieval. In Lecture Notes in Computer Science, SPIRE2012: 19th International Symposium on String Processing and Information Retrieval (Vol. 7608/2012, pp. 199 - 204). Cartagena de Indias, Colombia. October 21 - 25: Springer Berlin / Heidelberg.

[37] Dice, L. R. (1945, 07). Measures of the Amount of Ecologic Association between Species. In Ecological Society of America, 26, 297 - 302.

[38] Dong, A., Chang, Y., Zheng, Z., Mishne, G., Bai, J., Zhang, R., Diaz, F. (2010). Towards Recency Ranking in Web Search. In WSDM 2010: Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (pp. 11 - 20). New York, USA. February 3 – 6.: ACM Press.

[39] Dumais, S. T. (2005). Latent Semantic Analysis. In Annual Review of Information Science and Technology, 38(1), 188 - 230.

[40] Efron, M., & Golovchinsky, G. (2011). Estimation Methods for Ranking Recent Information. In SIGIR 2011: Proceedings of the 34th Annual International ACM SIGIR

Conference on Research and Development in Information Retrieval (pp. 495 - 504). Beijing, China. July 24 - 28.: ACM Press.

[41] Ferragina, P., & Gulli, A. (2008, 02). A Personalized Search Engine Based on Web-Snippet Hierarchical Clustering. In Software: Practice & Experience, 38(2), 189 – 225.

[42] Fleiss, J. L. (1971). Measuring Nominal Scale Agreement Among many Raters. In Psychological Bulletin, 76(5), 378 – 382.

[43] Freitag, D., Blume, M., Byrnes, J., Chow, E., Kapadia, S., Rohwer, R., & Wang, Z. (2005). New Experiments in Distributional Representations of Synonymy. In CoNLL 2005: Proceedings of the 9th Conference on Computational Natural Language Learning (pp. 25 - 32). Ann Arbor, Michigan. June 29 - 30.

[44] Ikehara, S., Murakami, J., & Kimoto, Y. (2003). Vector Space Model based on Semantic Attributes of Words. In JNLP: Journal of Natural Language Processing, 10(2), 111 - 128.

[45] Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. In Bulletin del la Société Vaudoise des Sciences Naturelles, 37, 547 - 579.

[46] Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. In TOIS: ACM Transactions on Information Systems, 20(4), 422 - 446.

[47] Jatowt, A., & Yeung, C. M. (2011). Extracting Collective Expectations about the Future from Large Text Collections. In CIKM 2011: Proceedings of the 20th ACM Conference on Information and Knowledge Management (pp. 1259 - 1264). Glasgow, Scotland, UK. October.: ACM Press.

[48] Jatowt, A., Kawai, H., Kanazawa, K., Tanaka, K., & Kunieda, K. (2009). Supporting Analysis of Future-Related Information in News Archives and the Web. In JCDL 2009: Proceedings of the Joint Conference on Digital Libraries (pp. 115 - 124). Austin, USA. June 15 - 19.: ACM Press.

[49] Jatowt, A., Kawai, H., Kanazawa, K., Tanaka, K., & Kunieda, K. (2010). Analyzing Collective View of Future, Time-referenced Events on the Web. In WWW 2010: Proceedings of the 19th International World Wide Web Conference (pp. 1123 - 1124). Raleigh, USA. April 26 – 30.: ACM Press.

[50] Joachims, T. (2002). Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms.: Kluwer/Springer.

[51] Jones, R., & Diaz, F. (2007). Temporal Profiles of Queries. In TOIS: ACM Transactions on Information Systems, 25(3). Article No.: 14.

[52] Kahle, B. (1997, 03). Preserving the Internet. In Scientific American Magazine, 276(3), pp. 72 - 73.

[53] Kanhabua, N., & Nørvåg, K. (2010). Determining Time of Queries for Re-Ranking Search Results. In ECDL 2010: Proceedings of The European Conference on Research and Advanced Technology for Digital Libraries. Glasgow, Scotland. September 6 – 10.: Springer Berlin / Heidelberg.

[54] Kanhabua, N., Blanco, R., & Matthews, M. (2011). Ranking Related News Predictions. In SIGIR 2011: Proceedings of the 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 755 - 764). Beijing, China. July 24 – 28.: ACM Press.

[55] Katzell, R. A., & Cureton, E. E. (1947). Biserial Correlation and Prediction. The Journal of Psychology, 24(2), 273 - 278.

[56] Kawai, H., Jatowt, A., Tanaka, K., Kunieda, K., & Yamada, K. (2010). ChronoSeeker: Search Engine for Future and Past Events. In ICUIMC 2010: Proceedings of the 4th International Conference on Ubiquitous Information Management and Communication (pp. 166 - 175). Suwon, Republic of Korea. January 14 – 15.: ACM Press.

[57] Li, X., & Croft, B. W. (2003). Time-Based Language Models. In CIKM 2003: Proceedings of the 12th International ACM Conference on Information and Knowledge Management (pp. 469 - 475). New Orleans, Louisiana, USA. November 2 – 8.: ACM Press.

[58] Liu, Y., Huang, X., An, A., & Yu, X. (2007). ARSA: A Sentiment-Aware Model for Predicting Sales Performance Using Blogs. In SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 607 - 614). Amsterdam, Netherlands. July 23 – 27.: ACM Press.

[59] Machado, D. (2009). Procura Estruturada de Textos para Perfis de Utilizadores. Master Thesis, Universidade da Beira Interior, Covilhã.

[60] Machado, D., Barbosa, T., Pais, S., Martins, B., & Dias, G. (2009). Universal Mobile Information Retrieval. In HCII 2009: Proceedings of the 13th International Conference on Human Computer Interaction (pp. 345 - 354). San Diego, USA. July 19 - 24.

[61] MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. In Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability (pp. 281 – 297).

[62] Mani, I., & Wilson, G. (2000). Robust Temporal Processing of News. In ACL2000: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (pp. 69 - 76). Hong Kong, China. October 1 – 8.: Association for Computational Linguistics.

[63] Matthews, M., Tolchinsky, P., Blanco, R., Atserias, J., Mika, P., & Zaragoza, H. (2010). Searching through time in the New York Times. In HCIR 2010: Proceedings of the 4th Workshop on Human-Computer Interaction and Information Retrieval (pp. 41 - 44). New Brunswick, USA. August 22.

[64] McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. In Psychometrika, 12(2), 153 - 157.

[65] Metzler, D., Jones, R., Peng, F., & Zhang, R. (2009). Improving Search Relevance for Implicitly Temporal Queries. In SIGIR 2009: Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 700 - 701). Boston, MA, USA. July 19 – 23.: ACM Press.

[66] Mishne, G., & Glance, N. (2006). Predicting Movie Sales from Blogger Sentiment. In CAAW 2006: Spring Symposium on Computational Approaches to Analysing Weblogs associated with AAAI2006: Twenty-First National Conference on Artificial Intelligence. Boston, Massachusetts, USA. July 16 - 20.

[67] Mori, M., Miura, T., & Shioya, I. (2006). Topic Detection and Tracking for News Web Pages. In WIC 2006: IEEE Main Conference Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (pp. 338 - 342). Hong Kong, China. December 18 – 22.: IEEE Computer Society Press.

[68] Nunes, S., Ribeiro, C., & David, G. (2007). Using Neighbors to Date Web Documents. In WIDM2007: Proceedings of the 9th ACM International Workshop on Web Information and Data Management associated with CIKM2007: 16th International Conference on Knowledge and Information Management (pp. 129 - 136). Lisboa, Portugal. November 9.: ACM Press.

[69] Nunes, S., Ribeiro, C., & David, G. (2008). Use of Temporal Expressions in Web Search. In C. Macdonald, I. Ounis, V. Plachouras, I. Ruthven, & R. White (Eds.),

Lecture Notes in Computer Science - Advances in Information Retrieval, ECIR 2008: European Conference on IR Research (Vol. 4956/2008, pp. 580 - 584). Glasgow, Scotland. 30th March - 3rd April.: Springer Berlin / Heidelberg.

[70] Osinski, S. l., Stefanowski, J., & Weiss, D. (2004). Lingo: Search Results Clustering Algorithm based on Singular Value Decomposition. In M. A. Klopotek, S. T. Wierzchon, & K. Trojanowski (Eds.), Intelligent Information Systems - Advances in Soft Computing, IIPWM2004: Intelligent Information Processing and Web Mining (pp. 359 - 368). Zakopane, Poland. May 17 – 20.: Springer Berlin / Heidelberg.

[71] Pantel, P., & Lin, D. (2002). Discovering Word Senses from Text. In KDD02: Proceedings of ACM Conference on Knowledge Discovery and Data Mining (pp. 613 - 619). Edmonton, Canada. July 23 - 26.

[72] Pearson, K. (1896). Mathematical contributions to the theory of evolution. III. Regression, heredity and panmixia. In Philos. Trans. Royal Soc. London Ser. A , 187 , 253 - 318.

[73] Pecina, P., & Schlesinger, P. (2006). Combining Association Measures for Collocation Extraction. In COLING/ACL 2006: Proceedings of the International Committee on Computational Linguistics and the Association for Computational Linguistics (pp. 651 - 658). Sydney, Australia. 17 - 21 July.

[74] Pustejovs.ky, J., Castaño, J., Ingria, R., Sauri, R., Gaizauskas, R., Setzer, A., & Katz, G. (2003). TimeML: Robust Specification of Event and Temporal Expression in Text. In IWCS2003: Proceedings of the 5th International Workshop on Computational Semantics (pp. 28 - 34). Tilburg, Netherlands. January 15 - 17.

[75] Pustejovs.ky, J., Verhagen, M., Sauri, R., Littman, J., Gaizauskas, R., Katz, G., Setzer, A. (2006). TimeBank 1.2. (L. D. Consortium, Ed.) Philadelphia, USA.

[76] Radinsky, K., Agichtein, E., Gabrilovich, E., & Markovitch, S. (2011). A Word at a Time: Computing Word Relatedness using Temporal Semantic Analysis. In WWW 2011: Proceedings of the 20th International World Wide Web Conference (pp. 337 - 346). Hyderabad, India. March 28 - April 1.

[77] Radinsky, K., Davidovich, S., & Markovitch, S. (2008). Predicting the News of Tomorrow Using Patterns in Web Search Queries. In WIC 2008: IEEE Main Conference Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (pp. 363 - 367). Sydney, Australia. December 9 – 12.: IEEE Computer Society Press.

[78] Scaiella, U., Ferragina, P., Marino, A., & Ciaramita, M. Topical Clustering of Search Results. In WSDM 2012: Proceedings of the 5th ACM International Conference on Web Search and Data Mining (pp 223 - 232). New York, USA. February 8 – 12.: ACM Press.

[79] Setzer, A., & Gaizauskas, R. (2000). Annotating Events and Temporal Information in Newswire Texts. In LREC2000: Proceedings of the 2nd International Conference on Language Resources and Evaluation. Athens, Greece. May 31 - June 2.: ELDA.

[80] Shaparenko, B., Caruana, R., Gehrke, J., & Joachims, T. (2005). Identifying Temporal Paterns and Key Players in Document Collections. In TDM 2005: Proceedings of the Workshop on Temporal Data Mining associated with ICDM2005 (pp. 165 - 174). Houston, USA. November 27 – 30.: IEEE Computer Society Press.

[81] Silva, J. F., Dias, G., Guilloré, S., & Pereira, J. G. (1999). Using LocalMaxs Algorithm for the Extraction of Contiguous and Non-contiguous Multiword Lexical Units. In EPIA 1999: Proceedings of the 9th Portuguese Conference in Artificial (pp. 21 - 24). Évora, Portugal. September 21 - 24.

[82] Song, R., Luo, Z., Nie, J.-Y., Yu, Y., & Hon, H.-W. (2009). Identification of Ambiguous Queries in Web Search. In Information Processing & Management: An International Journal, 45(2), 216 - 229.

[83] Strötgen, J., & Gertz, M. (2010). HeidelTime: High Quality Rule-based Extraction and Normalization of Temporal Expressions. In SemEval2010: Proceedings of the 5th International Workshop on Semantic Evaluation associated with ACL2010: 41th Annual Meeting of the Association for Computational Linguistics (pp. 321 - 324). Uppsala, Sweden. July 11 - 16.

[84] Strötgen, J., & Gertz, M. (2012). Multilingual and cross-domain temporal tagging. In LRE: Language Resources and Evaluation, 1 - 30.

[85] Strötgen, J., Alonso, O., & Gertz, M. (2012). Identification of Top Relevant Temporal Expressions in Documents. In TWAW 2012: Proceedings of the 2nd International Temporal Web Analytics Workshop associated with WWW2012: 21th International World Wide Web Conference (pp. 33 - 40). Lyon, France. April 17.: ACM Press.

[86] Turney, P. D. (2011). Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In EMCL 2001: Proceedings of the 12th European Conference on Machine Learning (pp. 491 - 502). Freiburg, Germany. September 5 - 7.

[87] Zamir, O., & Etzioni, O. (1998). Web Document Clustering: A Feasibility Demonstration. In SIGIR 1998: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 46 - 54). Melbourne, Australia. August 24 – 28.: ACM Press.

[88] Zhang, R., Chang, Y., Zheng, Z., Metzler, D., & Nie, J.-y. (2009). Search Result Re-ranking by Feedback Control Adjustment for Time-sensitive Query. In NAACL 2009: Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (pp. 165 - 168). Boulder, Colorado, USA. May 31 - June 5.